Modeling and simulation of multi-scale spiking neuronal networks

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Physiker
Maximilian Schmidt
aus Duisburg

Berichter: Universitätsprofessor Dr. Markus Diesmann
Universitätsprofessor Dr. Andreas Offenhäusser

Tag der mündlichen Prüfung: 24.05.2016

Zusammenfassung

Modellierung und Simulation vielskaliger spikender neuronaler Netze

Es ist ein zentrales Anliegen der Neurowissenschaften, den Zusammenhang zwischen Struktur und Dynamik in Netzwerken des Gehirns zu verstehen. Modelle von neuronalen Netzen sind nützliche Werkzeuge, um diesbezüglich Hypothesen zu überprüfen, weil sie nach Belieben modifiziert werden können. Die vorliegende Arbeit präsentiert ein Netzwerkmodell des visuellen Kortex eines Makaken, bestehend aus 32 kortikalen Arealen mit geschichteter Struktur und insgesamt mehr als vier Millionen Neuronen. Die Arbeit nimmt Probleme auf verschiedenen Ebenen des Prozesses der Modellierung und Simulation solcher Netzwerke in Angriff.

Software zur Simulation neuronaler Netze bedarf einer effizienten Repräsentation der Netzwerkkomponenten und muss die numerischen Gleichungen der Neuron- und Synapsenmodelle mit hoher Geschwindigkeit lösen, um die vorhandenen Rechenplattformen von Laptops bis zu Supercomputern optimal auszunutzen. Wir beschreiben hier die vierte Generation des Softwarekerns des Netzwerksimulators NEST und demonstrieren seine verbesserte Leistungsfähigkeit in Bezug auf die speichereffiziente Benutzung von Supercomputern. In Anbetracht der wachsenden Menge verfügbarer experimenteller Daten über die Struktur des Kortex dank neuer Technologien und koordinierter Anstrengungen großer Forschungsverbünde ist es eine Herausforderung, dieses Wissen in eine konsistente Modelldefinition auf der Ebene von Neuronen und Synapsen zu integrieren. Um das Netzwerkmodell zu konstruieren, führen wir eine große Sammlung von experimentellen Daten mit systematischen Gesetzmäßigkeiten kortikaler Struktur zu einer kohärenten Beschreibung der Anatomie zusammen. Unsicherheiten und Lücken in anatomischen Daten lassen dabei Spielraum für geeignete Netzwerkrealisationen, deren Bestimmung durch die hohe Dimensionalität solch vielskaliger Netzwerke erschwert wird. Wir präsentieren hier eine theoretische Methode, die es ermöglicht, experimentelle Aktivitätsdaten in die Modellkonstruktion einzubinden. Die Methode analysiert den Einfluss von strukturellen Komponenten des Modells auf seine Dynamik und bestimmt einen minimalen Satz von Modifikationen, die zu biologisch plausibler Aktivität führen. Daraus ergeben sich Hypothesen für zukünftige anatomische Studien und damit eine Rückwirkung von Netzwerkmodellen auf Experimente. Durch numerische Simulationen des Modells finden wir einen stabilen Grundzustand mit asynchroner, unregelmäßiger Aktivität und heterogenen Aktivitätsmustern in verschiedenen kortikalen Arealen. Die simulierte Aktivität weist Korrelationsmuster zwischen Hirnregionen auf, die denen von experimentellen Messungen im Makakenkortex ähneln.

Zusammengefasst etabliert diese Arbeit einen geschlossenen Workflow für die Konstruktion, Simulationen und Analyse großskaliger neuronaler Netzwerkmodelle. Das vorgestellte Modell reproduziert wichtige Aspekte kortikaler Aktivität und wird durch seine Erweiterbarkeit

auf verschiedenen Ebenen als Prüfstand für Hypothesen über die Beziehung von Struktur und Dynamik im Kortex dienen.

Summary

Modeling and simulation of multi-scale spiking neuronal networks

The link between structure and dynamics of brain networks is a key topic in neuroscience. Neuronal network models are useful tools to test corresponding hypotheses because they can be manipulated at will. This thesis presents a network model of macaque visual cortex comprising 32 cortical areas with laminar structure and a total of more than four million neurons. We address problems at different stages in the workflow of modeling and simulating such networks.

Neural network simulation software needs to efficiently represent the network components and solve the numerical equations of neuron and synapse models at high speed to make optimal use of computing platforms from laptops to supercomputers. We describe the fourth generation software kernel of the neural network simulator NEST and demonstrate its improved performance in terms of memory-efficient usage of supercomputing technology. With experimental data on cortical structure being increasingly available owing to new technologies and concerted efforts of large consortia, it remains an open challenge to integrate this knowledge into a consistent definition of a network model at the level of neurons and synapses. The construction of the network model presented in this thesis combines a large body of experimental knowledge with systematic regularities on cortical structure to arrive at a coherent anatomical description. Uncertainties and gaps in the data leave room for finding a good realization of the network connectivity, which is complicated by the high dimensionality of such multi-scale networks. We present a theoretical method which enables the integration of experimental activity data into the model construction. The method analyzes the impact of the structural components of the model on its collective dynamics and identifies a minimal set of modifications yielding biologically plausible activity. This leads to hypotheses for future anatomical studies and thus creates a feedback link from network models to experiments. Numerical simulations of the model reveal a stable ground state with asynchronous, irregular activity and heterogeneous activity patterns across cortical areas. The simulated activity displays a pattern of correlations between brain regions similar to that observed experimentally in macaque cortex.

In conclusion, this work establishes a closed-loop framework for the construction, simulation and analysis of large-scale neuronal network models. The presented model reproduces important aspects of cortical activity and, being amenable to refinements and extensions on multiple levels, will serve as a testbed for further hypotheses on the relationship between dynamics and structure in cortex.

Contents

I

Introduction

Focus of the study

The goal of neuroscience is to understand the functional principles of the brain and its components. Neuronal network simulations are helpful in this endeavor by describing parts of the brain with a system of mathematical models of neurons and synapses and reproducing experimental results on brain activity. Since models can be modified at will, e.g., by removing connections and changing parameters, they are useful tools for creating and testing hypotheses. Network models can be classified into two different categories. *Bottom-up models* start from a description of the model components and their connectivity and analyze the emerging dynamics. *Top-down studies*, on the other hand, define a desired functionality of the model, e.g., recognition of certain visual features, and then search for the network architecture necessary to achieve this.

The fundamental question underlying this work is **"How does the anatomical connectivity between neural populations shape cortical activity?"**. We approach this problem by constructing a bottom-up model of all vision-related areas of macaque cortex, combining simple single-neuron dynamics with complex structural connectivity and realistic anatomy. By neglecting details on the single-cell level, we bring out the influence of connectivity on the network dynamics.

Research on bottom-up models of cortex has hitherto followed two different streams. On the one hand, network models of local circuits incorporate varying degrees of detail on the single-cell level and on local connectivity. These studies investigate aspects of cortical dynamics such as statistical properties of spiking neurons (Traub et al., 2005; Haeusler et al., 2009), slow oscillations arising in thalamo-cortical circuits (Hill & Tononi, 2005), the response to external stimuli (Rasch et al., 2011), and the impact of local connectivity structure on network dynamics (Haeusler et al., 2009; Voges & Perrinet, 2012). On the other hand, large-scale network models simulate the global dynamics of cortex and represent single areas or populations with simplified models. For instance, areas are modeled as oscillators using the Wilson-Cowan model (Deco et al., 2009) or Kuramoto oscillators (Cabral et al., 2011), or as small, highly simplified networks of spiking neurons (Deco & Jirsa, 2012). These studies have been able to explain certain features like the anti-correlated activation of groups of areas in the resting-state condition (Fox et al., 2005; Fox & Raichle, 2007). Sec. 1.8 elaborates on previous work in the field.

Both types of models have their limitations. A cortical neuron receives approximately half of its inputs from neighboring cells while the other half stems from more distant neurons inside its area, from other cortical areas, and from non-cortical brain regions. It is thus both integrated in a local circuit and embedded in a global network. Consequently, cortical function is the result of interactions on different scales from local processing inside areas to communication between

distant areas, involving feedforward and feedback processes (Lamme et al., 1998; Pascual-Leone & Walsh, 2001; Riesenhuber & Poggio, 1999; Rao & Ballard, 1999; Bastos et al., 2015a). Our model of macaque visual cortex bridges these scales by including both local structure of cortical areas and long-range projections between them.

Three major obstacles at all stages of the modeling workflow (Fig. 1) have impeded the simulation of large-scale spiking networks and are now gradually being overcome. First, simulating large-scale spiking networks requires a large amount of memory to represent all neurons and synapses, and intense communication in parallel simulations, leading to high demands on available hardware and software. Prior to this work, the network simulator NEST (NEural Simulation Tool) was already shown to scale well up to 10^5 compute cores on the supercomputers JUQUEEN (Juelich, Germany) and K (Kobe, Japan), enabling the simulation of networks of up to 10^8 neurons (Helias et al., 2012) with a realistic indegree of $\sim 10^4$ incoming synapses per neuron. Constituting a substantial part of cortex, the visual cortex exceeds these numbers by far, with estimates for the size of the entire macaque cortex in one hemisphere ranging from $8 \cdot 10^8$ (Gabi et al., 2010) to $29.5 \cdot 10^8$ (Lidow & Song, 2001) neurons. A full simulation thus requires the improvement of NEST's internal representation of the network structure to efficiently exploit larger machines.

Second, knowledge on cortical structure and connectivity has been sparse and scattered across many anatomical studies. The CoCoMac database (Stephan et al., 2001; Bakker et al., 2012) collects results of hundreds of axonal tracing studies in the macaque monkey, facilitating the usage of such data to compile network connectivities. Recent works by Markov et al. (2014a,b) report quantitative results on connection densities and laminar patterns of cortico-cortical connections. Nonetheless, data is still sparse and gaps have to be filled using statistical regularities on cortical structure. For instance, Ercsey-Ravasz et al. (2013) found connection densities in cortex to decay with the distance between areas.

Third, large-scale neural networks are high-dimensional dynamical systems with a vast parameter space. In the purely forward modeling approach, sketched by the black arrows in Fig. 1, parameters are usually tuned manually to obtain satisfactory activity, which becomes unfeasible for high-dimensional networks. The relationship between model structure and dynamics is a priori unknown, so that finding a minimal set of modifications to improve the model activity is difficult. Therefore, we need to shift the perspective and additionally consider the system in a reverse manner, predicting the structure necessary to explain the observed activity (red arrows in Fig. 1).

In this work, we address all three problems and achieve the construction of the multi-scale spiking network model of macaque visual cortex, which we analyze with regards to experimental

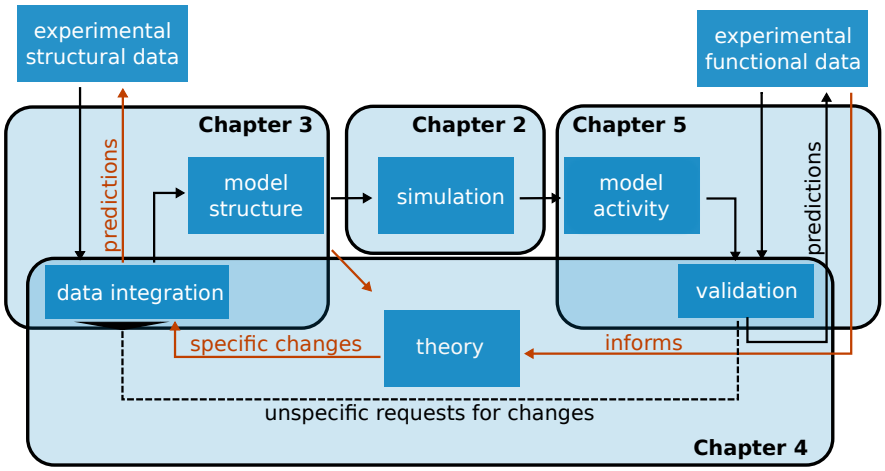

Figure 1: Workflow of bottom-up network simulations. Figure adapted from Schuecker et al. (2015b) with permission. Experimental structural data is integrated to define the structure of the model. Together with a specification of the single-neuron and synapse dynamics, it yields a certain model activity via numerical simulation. The resulting activity is validated with experimental functional data. This comparison either leads to predictions for future experiments or, in case of disagreement, leads to unspecific requests for changes in the model. Informed by experiments, the analytical theory presented in Chap. 4 rather identifies specific changes improving the model and thereby creates predictions for structural experiments. The chapters of this thesis address issues in all stages of the workflow.

findings. The thesis is structured as follows: In the remainder of Part I, the scientific background of this work is explained, with an introduction to the mammalian cortex and to biological neurons. Furthermore, basics of computational neuroscience are introduced including single-cell models, balanced random networks and the analytical theory of such networks. We summarize the main findings of the model of Potjans & Diesmann (2014) simulating a $1\,\mathrm{mm}^2$ patch of early sensory cortex, which forms the basis of our multi-area model, and review previous work on cortical models. Finally, we introduce the network simulator NEST.

In Part II, the results of this study are presented. In Chap. 2, we identify points for improvement in the NEST kernel, describe the development of the fourth generation NEST kernel with a new memory infrastructure, and assess its performance by scaling benchmarks. Chap. 3 presents the integration of a large body of experimental knowledge into a consistent definition of a multi-scale model of visual cortex and details the regularities used to bridge gaps in the available data. Chap. 4 describes the theory allowing us to identify connections in the multi-area model critical to its dynamics to find a specific set of modifications necessary to align the model with experimental activity data. Using this approach, we extend purely structure-based model definitions to include activity constraints and generate predictions on the structure of the brain. In Chap. 5, we demonstrate that the resulting network reproduces a realistic ground state with asynchronous, irregular spiking activity and heterogeneous firing rates across populations and layers. The model features increased intrinsic time scales in single-cell spiking activity of higher compared to early visual areas as well as cortico-cortical correlation patterns

similar to those measured by functional magnetic resonance imaging. Finally, we discuss our findings in Part III and give an outlook on future projects. The supplementary material in Chap. 6 contains more detailed derivations of the basic equations of single-cell models and the analytical theory as well as additional material on the results chapters.

CHAPTER 1

Fundamentals

This chapter provides the biological and theoretical background for the results section. It focuses on macaque cortex. However, most of the findings on structure and dynamics have been found or are thought to hold for other mammals like the human as well.

1.1 Neurons and synapses

The macaque visual cortex comprises on the order of 10^9 neurons across two hemispheres (Gabi et al., 2010; Lidow & Song, 2001) and these cells differ vastly in their morphological structure, dynamical behavior and functional tasks. Dale's principle states that a neuron is either excitatory or inhibitory, i.e., it either increases or decreases the membrane potential of postsynaptic neurons. A neuron consists of four basic structural components. The *soma* or cell body contains the cell nucleus and maintains a potential difference between the interior of the cell and its outer environment, the membrane potential, on the order of $-70\,\mathrm{mV}$ at rest (Fig. 1.1). The cell body integrates action potentials which the neuron receives via the synapses, mainly on its *apical* and *basal dendrites*. A *chemical synapse* is a contact between the *axon* of a presynaptic neuron and the postsynaptic neuron with the synaptic cleft in between. The contact is formed with a *dendritic spine*, a small protrusion on the dendrite, or directly with the dendritic shaft, the axon or the soma of the postsynaptic partner. On the arrival of an *action potential*, also called *spike*, ion channels open to alter the ion concentration in the axon terminal causing vesicles to release neurotransmitters into the synaptic cleft where they bind to postsynaptic receptors (Fig. 1.1). The receptors open channels which provide ions including sodium (Na^+), potassium (K^+), chloride (Cl^-), and calcium (Ca^{2+}) causing a positive or negative membrane potential deflection to propagate to the cell body. The *synaptic weight* describes the amount of membrane potential change induced by the arrival of a spike at the synapse. In computational neuroscience, the synaptic weight is often described by a single number, while in biology, it depends on a number of different factors, such as how well the signal propagates through the postsynaptic dendrite, the release probability of the vesicles and the amount of neurotransmitter released into the synaptic cleft, typically correlated with the size of the dendritic spine. Chemical synapses feature a rich repertoire of different *plasticity* dynamics, varying the synaptic weight on different time scales from hundreds of milliseconds in short-term plasticity to minutes and hours in homeostasis. For instance, synapses with *spike-timing dependent plasticity* change their weight based on the relative timing between spikes emitted by the presynaptic and postsynaptic partners. Once the membrane potential has reached a certain threshold level, its ion channels open and a spike is emitted. While chemical synapses are the major gates of communication between neurons, cells can also connect via electrical synapses, so-called gap junctions, where two cell bodies are in close apposition, allowing for direct transmission of electrical impulses between the cells.

The majority of excitatory neurons in the cerebral cortex are *pyramidal neurons*, which have a characteristic triangular cell body (Fig. 1.1) and dendrites with a large spread across the cortical tissue. A common type of inhibitory neuron is the *stellate cell*, whose name stems

from the characteristic star-like shape of its dendrites (Fig. 1.1). Similar to synapses, neurons exhibit different types of internal dynamics. They can be distinguished based on their response to standard stimulation protocols, e.g., how they adapt to imposed stimulation frequencies or whether they intrinsically tend to emit spikes at irregular intervals. For instance, Markram et al. (2015) found 55 distinct morphological types of neurons and 11 different dynamical types in a column of somatosensory cortex of rat (Fig. 1.2). Combining them led to the identification of 207 different types of neurons in the brain tissue under study. The following sections show that different activity patterns can, however, also be explained by network effects. Neurons in the brain thus generate their rich repertoire of dynamics both from their intrinsic properties and from the network in which they are embedded.

1.2 Structure of macaque cortex

This section gives a brief introduction to the anatomy of the macaque cortex to set a basis for the understanding of the multi-area model of macaque visual cortex. It does not aim to provide a complete review of current knowledge but to convey a general overview.

The cerebral cortex, often referred to as the gray matter, is the outermost structure of the brain of humans and other mammals (Fig. 1.3). It consists mainly of cell bodies and capillaries in contrast to the underlying white matter which mostly contains axons of neurons. Its folded structure developed during evolution and enables more cells to fit in the restricted volume of the skull. Cortex shows a universal pattern of sulci and gyri, whose exact locations are specific to each individual and, among other anatomical landmarks, are defined in an *atlas* of the brain. To compare findings from different individuals, it is necessary to map their coordinates onto a standard surface, so that the different gyri and sulci coincide. For the macaque, one example is the F99 space, a standard macaque cortical surface included with Caret, a software for surface mapping purposes (Van Essen et al., 2001).

The macaque cortex can be parcellated into *areas* based on different criteria that can be functional ("What is the dynamical behavior of a cortical region?") or structural ("What are anatomical properties of the region?"). Areas are associated with one or more senses of perception, for instance sight, hearing or touch, or cognitive function such as memory or language processing. Felleman & Van Essen (1991) derived a parcellation scheme of macaque visual cortex, abbreviated as FV91 scheme, which is used for the model defined in Chap. 3. They used three methodological approaches to identify different areas. *Connectivity analysis* exploits the characteristic patterns of incoming and outgoing connections of each cortical area and mainly relies on tracing studies (see below). *Architectonics* distinguishes areas by their distinct structure, such as cell densities, which are observed using different staining techniques. Moreover,

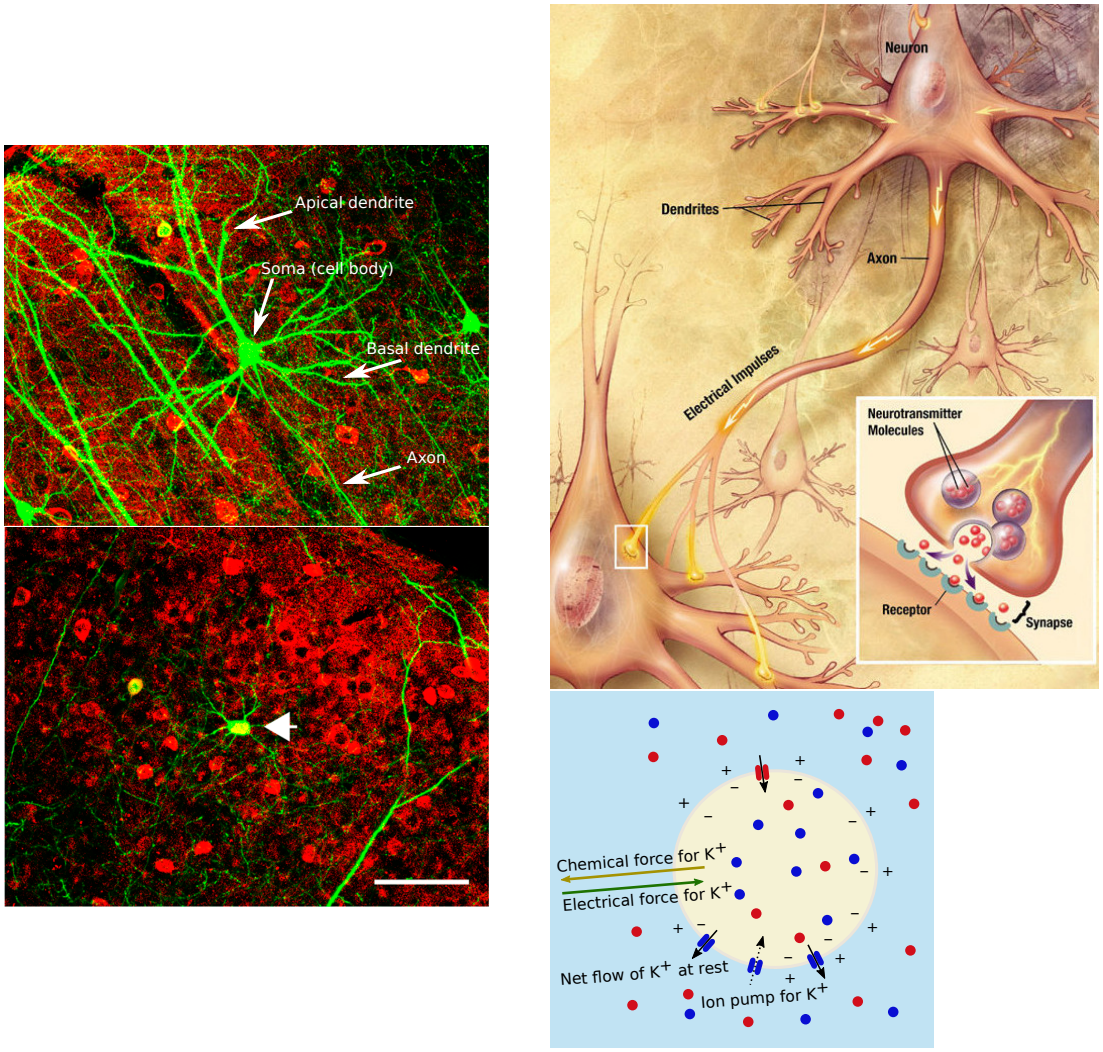

Figure 1.1: Biological neurons. Top left: Pyramidal cell in mouse visual cortex. **Bottom left**: Non-pyramidal cell stained in mouse visual cortex. Both figures adapted from Lee et al. (2005) with permission. **Top right:** Sketch of a chemical synapse (source: National Institute of Aging, National Institutes of Health, http://www.nia.nih.gov/alzheimers/publication/alzheimers-disease-unraveling-mystery/preface, in public domain.) **Bottom right:** Cell body of neuron at resting potential. The cell establishes a potential difference between intracellular (yellow) and extracellular (blue) space by maintaining the concentration gradients of the different ion species. Each ion species is subject to a chemical force caused by the concentration gradient and an electrical force which result in an individual reversal potential at which the ion flow vanishes. The resting potential is a weighted average of the reversal potentials of all four ion species (Na^+, K^+, Cl^-, and Ca^{2+}) where Na^+ (red circles) and K^+ (blue circles) are most influential. At rest, there are resulting inward and outward currents of the different ions through ion channels in the cell membrane which are counteracted by ion pumps.

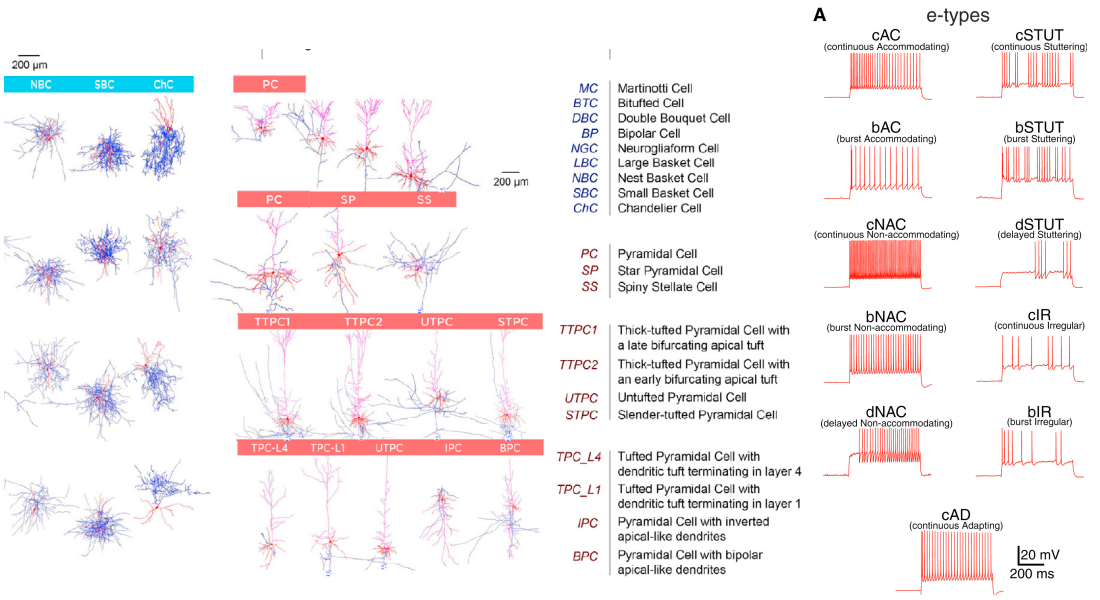

Figure 1.2: Diversity of cortical neurons. Findings from anatomical and electrophysiological experiments in rat somatosensory cortex. **Left:** Morphological types of excitatory and inhibitory neurons (with blue headlines). Dendrites in red, axon in blue. **Right:** Subset of the different electrical types of neurons, characterized by their response to depolarizing step current injections. Excitatory neurons are all *continuous accomodating* neurons, and inhibitory neurons can be of any of the ten remaining types. Figure adapted from Markram et al. (2015), with permission.

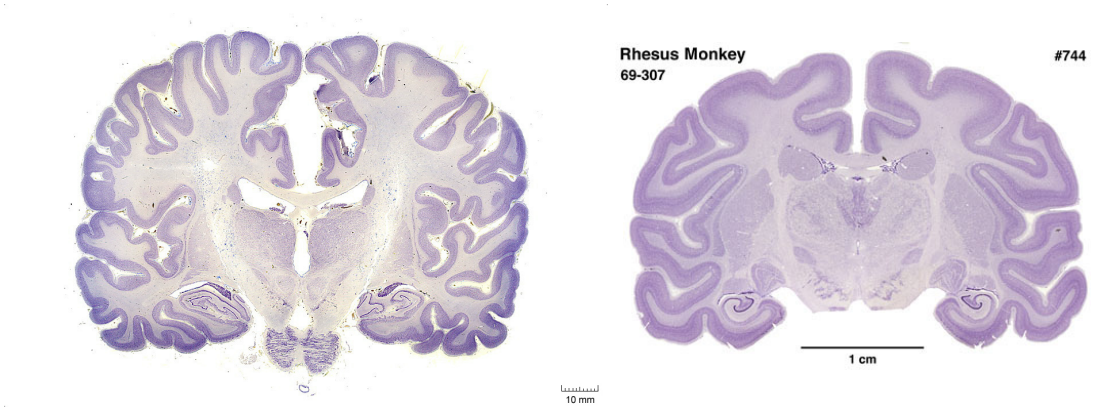

Figure 1.3: Comparison of human and macaque cerebral cortex. Nissl stained sections of human brain (left) and rhesus macaque brain (right). The cortex is the outer layer colored in deep purple. The white matter is the light-colored structure underneath the cortex. Both figures reproduced with permission from http://www.brains.rad.msu.edu, and http://brainmuseum.org, supported by the US National Science Foundation.

each area has its individual *topographic organization*. For instance, in primary visual cortex (V1), different regions show individual responses to specific grating orientations, measurable in physiological experiments, and such *receptive fields* are differently organized across areas. In addition, specific physiological characteristics of neurons and behavioral consequences of lesions in parts of cortex were used to identify areas. Fig. 1.4A shows the resulting parcellation of the macaque visual cortex into 32 areas on a flat map of cortex.

Cortex has a layered structure with layers differing in the cell types they contain, their neuronal density, and their connectivity structure (Sec. 1.6). Based on their degree of lamination and the thickness of the granular layer 4, areas are distinguished into eight different architectural types (Barbas & Rempel-Clower, 1997; Dombrowski et al., 2001). The vision-related areas fall into categories 2, 4, 5, 6, 7 and 8, of which four are shown in Fig. 1.4. Agranular cortex does not contain a layer 4. Among vision-related areas, only parahippocampal area TH is of this type. Primary visual cortex V1, in contrast, is a eulaminate area and contains six distinct layers, and even a finer distinction is possible. Layer 1 is the main input layer for cortico-cortical feedback connections and contains many synapses, but few cell bodies. It is thus often only implicitly represented in computational models. The granular layer 4 and infragranular layer 6 are the main input layers for signals from the thalamus, a subcortical structure involved in the visual pathway between retina and cortex. After processing the information, the supragranular layers 2 and 3, which are often regarded as one layer 2/3, are the main output layers forwarding the signal to higher cortical areas.

The connectivity within and between areas has been measured with various methods, complementing each other on different scales. On very small scales of nanometers, microscopic investigations of brain slices are used to reconstruct the complete connectivity graph between individual neurons and recent progress has been made to automate and speed up the process (Helmstaedter, 2013; Kasthuri et al., 2015; Berning et al., 2015). A promising recently developed technique is CLARITY, where brain tissue is made transparent post mortem (Chung & Deisseroth, 2013). When accompanied by gene-based labeling, the method enables highly detailed pictures of specific pathways in the brain. Binzegger et al. (2004) reconstructed neurons in slices from cat primary visual cortex and combined measurements on their dendritic and axonal lengths (Sec. 1.1) with a modified version of *Peters's rule* (Peters & Feldman, 1976; Braitenberg & Schüz, 1998). Peters's rule states that the density of synaptic contacts between two neurons is proportional to the overlap of their dendritic and axonal trees where the proportionality constant may depend on the cell types but not on the individual neurons or axonal or dendritic branches. A different approach is to probe connections by stimulating a neuron and measuring the response in potentially connected neurons (see Thomson & Lamy (2007) for a review). On a larger scale, connections between brain areas have been extensively studied by

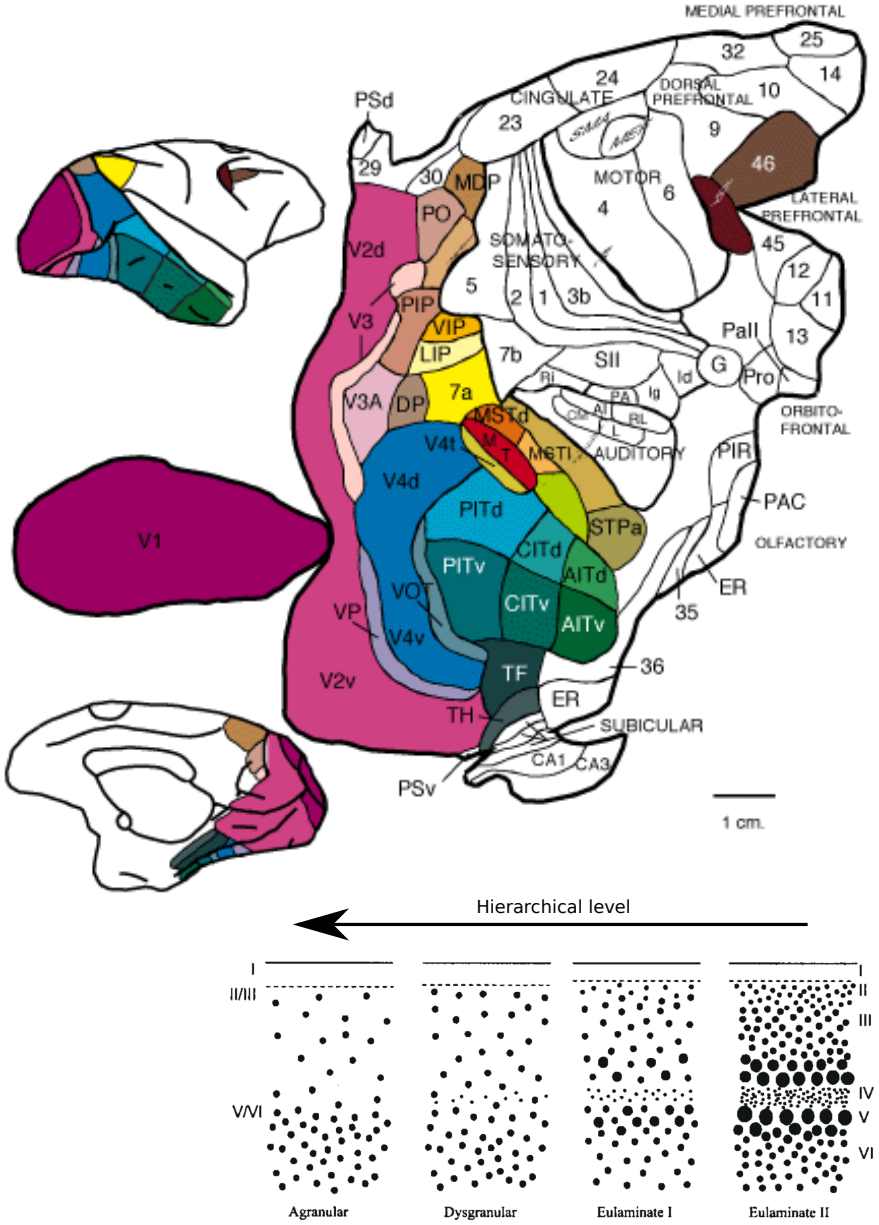

Figure 1.4: Areas of the visual cortex. Top left: Flat cortical map adapted with permission from Felleman & Van Essen (1991). Vision-related areas are colored according to the lobes in which they are located: Occipital lobe in purple, blue, and reddish hues, parietal lobe in yellow, orange, and light brown, temporal lobe in shades of green and frontal lobe in dark shades of brown. **Top right**: Hierarchy of the visual cortex adapted with permission from Felleman & Van Essen (1991) from early visual stages (bottom) to higher associative areas (top). Coloring as in the left panel. **Bottom**: Neuron densities in cortical areas of different architectural types. Figure adapted with permission from Fig. 10 in Dombrowski et al. (2001).

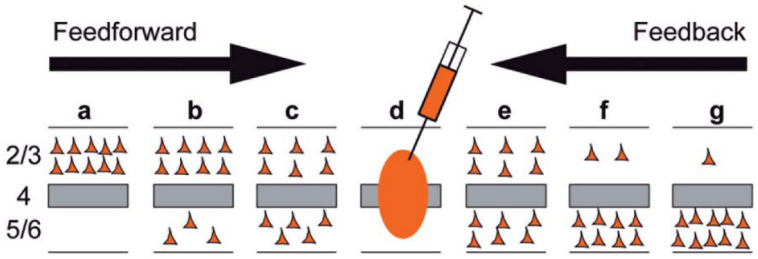

Figure 1.5: Retrograde tracing experiment. Figure adapted with permission from Markov et al. (2014b). A retrograde tracer is injected into area d, is taken up by axon terminals and from there propagates along the axon until it labels a cell body. The relative number of labeled neurons in a projecting area c gives the connection strength between c and d. Moreover, the fraction of supragranular (layer 2/3) labeled neurons in area c gives information about the preferred laminar origin of the cortico-cortical connection.

tracer injections. To summarize knowledge about brain connectivity from tracing experiments, the CoCoMac database (Stephan et al., 2001; Bakker et al., 2012) collects data from over 400 studies. It contains connection densities on a rough discrete scale. Markov et al. (2014a) introduced a more graded scale by counting numbers of labeled neurons projecting to areas injected with retrograde tracers (Fig. 1.5). Moreover, they quantitatively measured laminar connection patterns (Markov et al., 2014b). Another technique is polarized light imaging (PLI) of sliced brain tissue to measure the orientation of fibers in the white matter exploiting their optical refraction of polarized light (Axer et al., 2011). This technique requires tractography algorithms to construct connectivity matrices which are prone to be unreliable over large distances and are sensitive to the parameters chosen. Moreover, the alignment of brain slices is challenging and tends to introduce inaccuracies.

The downside of all these methods is that they are invasive and require the sacrifice of the subject. Diffusion tensor imaging (DTI) is a new, non-invasive method which measures the orientation of nerve fibers, exploiting the fact that the diffusion of water molecules is higher along nerve fibers than orthogonal to them (Hagmann et al., 2007). From this information, one reconstructs the density of fibers connecting a particular pair of voxels, using sophisticated algorithms. While DSI in principle allows one to measure the connectivity between all pairs of brain areas it cannot detect the direction of a connection and cannot compete with tracing data in terms of accuracy and reliability (Thomas et al., 2014).

The cortico-cortical connectivity graph is dense at the area level, e.g., Markov et al. (2014a) found that 66 % of all possible connections to 29 cortical areas are realized, while connection densities vary across five orders of magnitude (Markov et al., 2014a). Based on a set of connection rules, Felleman & Van Essen (1991) compiled a hierarchy of visual areas which reflects functional relationships in the sense that the early visual areas perform relatively simple calculations and higher areas process complex features and interact with other cortices. As a

general rule, *feedforward connections*, i.e., pathways from lower to higher hierarchical areas, are preferentially initiated by superficial cells which establish synapses in the granular layer 4 of the target area. In the other direction, *feedback connections* start mainly in the infragranular layers 5 and 6 and form synapses in all layers except layer 4. Using quantitative data has led to alternative hierarchies with continuous scales (Barone et al., 2000; Reid et al., 2009; Markov et al., 2014b), which are, however, still in overall agreement with the original hierarchy of Felleman & Van Essen (1991). As sketched in Fig. 1.4, hierarchical rank approximately corresponds to the architectural type of an area, e.g., areas tend to have lower cell densities and decreasingly clear lamination with increasing hierarchical level.

There are two functional streams in visual cortex (for instance reviewed by Nassi & Callaway, 2009). The *dorsal stream* comprises areas in the parietal and occipital lobes, which process information about locations and movements (answering the question *"Where?"*) in the visual field. Areas in the *ventral stream* are positioned in the occipital and temporal lobes and determine the nature of objects (*"What?"*). Early visual areas V1 and V2 process basic information and are not assigned to either stream. Frontal areas 46 and FEF are integrative areas which interact with other modalities of cortex such as auditory or motor cortex. The two streams can also be distinguished based on their cortical connectivity, as Chap. 5 shows. Markov et al. (2013) found that the connectivity graph follows a *rich-club structure* with a central group of highly connected areas and two peripheral groups which are more weakly connected and form feedforward and feedback streams with the central group. These considerations show that the macaque cortex is a complex network with a multi-scale structure following different organizational principles on each scale.

1.3 Cortical dynamics

This section gives an introduction to the dynamics of cortex, restricted to *resting-state* dynamics, in which the subject does not perform a task. Single-cell activity is recorded by implanting electrodes in the intact brain (*in vivo*), using either single electrodes or electrode arrays which record from $\sim 10^2$ neurons. In general, neurons exhibit spike rates in the range between 0.1 and 30 spikes/s (Swadlow, 1988; de Kock & Sakmann, 2009). Inhibitory neurons generally show higher activity than excitatory cells (Swadlow, 1988; Sakata & Harris, 2009). A number of studies dealt with laminar patterns of spike rates in early cortical areas of mouse, rat and rabbit. They consistently reported that the excitatory populations in different cortical layers can be ordered as 2/3, 6, 4, and 5 with regard to their average spike rates with lowest rates in 2/3 and highest rates in 5, in a range from 0.1 to 10 spikes/s (see Table 6 of Potjans & Diesmann, 2014).

Single neurons generally exhibit high variability of single spike timing and emit spikes at irregular intervals. While in the primary visual cortex, neuronal spiking activity is often approximated as a Poisson process, the degree of irregularity varies a lot across cortical areas. Bernacchia et al. (2011) recorded single-cell activity from three areas in macaque cortex in a game task and studied how long a reward in the past influences the activity of a neuron. They found that time scales in different neurons ranged from hundreds of milliseconds to tens of seconds. Murray et al. (2014) studied the autocorrelation function of single-cell spiking activity and found that they decay on different time scales across cortical areas. These time scales increase with visual hierarchy, indicating that different functional properties of cortical areas are also reflected in the intrinsic spiking activity of neurons.

Correlations between spike trains of cortical neurons are low. For example, Ecker et al. (2010) measured activity in neighboring V1 cells and found very low correlations although these neurons share a significant amount of common input, due to the high local connectivity. Consequently, the activity of these neurons must be actively decorrelated, and theoretical works show that inhibitory feedback actively suppresses pairwise correlations in neuronal activity (Tetzlaff et al., 2012; Helias et al., 2014). Though many hypotheses on neural coding assume groups of neurons to encode stimuli with firing rates and thus rely on correlated activity, most of these theories require an asynchronous ground state of the network, i.e., with low noise correlations, from which the stimulus-evoked correlations can arise. Moreover, uncorrelated activity can be beneficial because the information that can be read out from a group of spike trains is maximal if the spike trains span an orthogonal basis, i.e., if the spiking activity is uncorrelated (Tripp & Eliasmith, 2007; Averbeck et al., 2006).

Experimental measurements of single-cell spiking activity usually cover a few to maximally 10^2 neurons simultaneously, while cortical networks contain on the order of $10^5 - 10^8$ cells. Electrophysiological experiments therefore heavily undersample neural activity. Furthermore, determining population rates from measurements of single-cell spiking activity is controversially debated because of the "Dark Matter of Cortex"-problem (Shoham et al., 2006). This states that in single electrodes the number of recorded neurons is lower than one would expect from combining cell densities and the spatial reach of the electrodes, suggesting that many neurons go undetected. Consequently, findings from such experiments only provide a rough indication of cortical activity.

A complementary observable is the *local field potential* (LFP), which reflects synaptic activity from many neurons and can have a spatial reach of up to a few millimeters depending on the correlation of neuronal activity (Lindén et al., 2011). It corresponds to the low-frequency component ($\lesssim$ 500 Hz) of the electrical signal recorded by an extracellular electrode and is used to investigate ongoing and evoked oscillations in neural activity. LFP and spike rate

spectra have a dominant part in the low-frequency range, showing that despite small correlations between the activity of different neurons, there are ongoing oscillations in cortex (Belitski et al., 2008). Such oscillations are hypothesized to play a crucial role for the communication between areas. Fries (2005) suggested that inter-area oscillations open and close time windows for communication by controlling neuronal excitability during their peaks and troughs, respectively, known as *communication through coherence*. The original hypothesis was refined by Bastos et al. (2015b) to include the recent finding on laminar patterns of cortico-cortical connectivity (Markov et al., 2014b) and suggests that feedforward and feedback connections communicate on different oscillation frequencies. This is supported by recent findings, for instance, van Kerkoerle et al. (2014) observed early visual areas V1 and V4 to communicate in feedforward direction with fast oscillations ($\sim 55-65$ Hz) and in feedback direction with intermediate oscillations ($\sim 8-12$ Hz). Bastos et al. (2015a) studied 28 pairs of visual areas and consistently found that feedforward communication is carried out with slow (~ 4 Hz) and fast ($\sim 60-80$ Hz) oscillations, while feedback influences use synchronization on intermediate time scales ($\sim 14-18$ Hz).

On the brain scale, functional magnetic resonance imaging (fMRI) measures the time-dependent level of oxygen in the blood supporting brain cells, which is correlated with the collective activity of large groups of neurons. Its spatial resolution can be up to tens of micrometers, but this comes at the cost of low temporal resolution on the order of seconds. Recordings have revealed that even without external stimulation, cortical areas interact with each other in *resting-state networks* both in human (Fox et al., 2005; Fox & Raichle, 2007) and macaque cortex (Vincent et al., 2007; Shen et al., 2012, 2015). These are alternating groups of areas with internally correlated activity, measured via the zero-lag cross-correlation, called *functional connectivity*, of fMRI time series. In human visual cortex, resting-state networks distinguish dorsal and ventral attention systems (Fox et al., 2006) involving several visual areas (cf. Sec. 1.2). Vincent et al. (2007) demonstrated the emergence of correlated sets of areas in anesthetized macaque monkeys such as the oculomotor network including FV91 visual areas FEF, LIP, MST, and MT, the foveal and peripheral network including V1, V2 and MT; and a network resembling the so-called default mode network in human brain. These networks reflect the underlying structural connectivity, but also rely on dynamical effects. Deco et al. (2011) hypothesized that their alternating activation reflects the brain exploring its internal states to be better prepared for the presentation of an external stimulus. While multiple computational studies have investigated large-scale resting-state dynamics with simplified models of brain areas (Cabral et al., 2011; Deco & Jirsa, 2012), it remains an open question how these large-scale interactions relate to internal dynamics of the areas.

1.4 The leaky integrate-and-fire neuron model

Computational models of biological neurons vary in their complexity and scope. While multi-compartment neuron models take the spatially extended geometry of cells into account, point-neuron models reduce the cell with its axon and dendrites to a point in space. The leaky integrate-and-fire (LIF) neuron (Lapicque, 1907; Stein, 1967) is a simple point-neuron model which is widely used because it is analytically tractable while still being able to reproduce in vivo-like activity patterns. Other models consider the complex dynamics of ion channels, such as the Hodgkin-Huxley model (Hodgkin & Huxley, 1952).

Here, we model synaptic input as exponentially decaying currents injected into the soma of the neuron. Another possibility is to treat incoming spikes as changes in the cell body's conductance. This produces a more complex, sometimes more realistic behavior, which is more difficult to treat analytically. It is known that conductance changes evoked by excitatory inputs are mostly local to the synapse (Araya et al., 2006b), with limited influence on the somatic conductance. As a result, excitatory inputs on dendritic spines can sum linearly even when they are on the same branch (Araya et al., 2006a). However, distal inputs are subject to nonlinear amplification (Branco & Häusser, 2011). The influence of inhibitory inputs depends on their location (Jadi et al., 2012) either directly at the cell body or on dendritic shafts, where they reduce each other's effect if located on the same dendritic branch (Araya et al., 2006a). Neglecting such details about inhibitory inputs, one can treat synaptic input as current changes when modeling the cell as a point in space.

The LIF neuron models a biological neuron as a circuit of a resistance and a capacitance connected in parallel (Fig. 1.6A), where the resistance replaces the voltage-gated ion channels of the cell membrane and the capacitance mimics the membrane's ability to store electrically charged ions. The membrane potential V and the synaptic input I_s are described by two differential equations reading

$$\begin{aligned} \tau_\mathrm{m}\frac{\mathrm{d}V}{\mathrm{d}t} &= -(V - E_\mathrm{L}) + \frac{\tau_\mathrm{m}}{C_\mathrm{m}} I_\mathrm{s}(t) \\ \tau_\mathrm{s}\frac{\mathrm{d}I_\mathrm{s}}{\mathrm{d}t} &= -I_\mathrm{s} + \sum_i \delta(t - t_i) J\,, \end{aligned} \tag{1.1}$$

where τ_m is the membrane time constant, C_m is the capacitance of the neuron, E_L is the leak potential, τ_s is the synaptic time constant and J is the maximum of the post-synaptic current induced by a single spike. Throughout this thesis, we will set the single-neuron parameters to the value given in Table 1.1.

The arrival of a single spike evokes a post-synaptic potential (PSP) in the receiving neuron

which has the form of the difference between a fast-rising exponential with time constant τ_s and a slowly decaying exponential with time constant τ_m (Fig. 1.6A, Sec. 6.1). Eventually, the membrane potential decays back to the leak potential. The membrane potential follows the linear equation in Eq. 1.1 and emits a spike when it crosses the fixed threshold θ. Then it is manually set to the reset potential V_r and clamped to $V = V_r$ for an absolute refractory period τ_r. The asymptotic free membrane potential V_∞ of the neuron is the potential it would reach for $t \to \infty$ if the threshold were set to infinity. Furthermore, we define μ_{th} as the current that would asymptotically drive the neuron to θ Eq. 6.11.

Depending on the nature of the input, the neuron will display different types of behavior: For a constant input current ($I_s(t) = \mu$) with zero fluctuations ($\sigma = 0$), the neuron is in the *mean-driven regime* and the time course of the sub-threshold membrane potential follows the charging curve of an RC circuit (see Eq. 6.9). It will either settle in an equilibrium $V_\infty < \theta$ ($\mu < \mu_{th}$) or, if $\mu > \mu_{th}$ and thus $V_\infty > \theta$, it will emit spikes at regular intervals (Fig. 1.6C). For a noisy input current with mean $\mu < \mu_{th}$ and $\sigma > 0$, the neuron enters the *fluctuation-driven regime* (Fig. 1.6D). The fluctuations in the noisy input occasionally drive the neuron above threshold leading to an irregular series of spikes and fluctuations of the membrane potential. The resulting spiking rate depends both on μ and σ. This situation is particularly interesting for the use in models of the brain because it resembles the in vivo condition.

The F-I-curve measures the spike rate F of the neuron in response to an input current I with mean μ and standard deviation σ. Fig. 1.6B illustrates the crucial role of fluctuations in the input: For a DC current ($\sigma = 0$) as in Fig. 1.6C, μ_{th} marks a sharp border, where the input current drives the neuron above the threshold ($V_\infty > \theta$) and the neuron starts to spike. For increasing variance, this transition is softened and the neuron is driven above threshold even for $\mu < \mu_{th}$, i.e., the fluctuations in the noisy input occasionally cause the neuron to spike. For very high mean inputs, all curves converge to the same resulting firing rate, regardless of the strength of fluctuations. In the fluctuation-driven regime, the spike train of the neuron is often approximated as the result of a Poisson point process with its rate parameter set to the neuron's firing rate. In the present work, we use this approximation to derive an analytical mean-field theory describing the stationary state of spiking networks.

1.5 Balanced random networks

A *balanced random network* (BRN) is composed of populations of excitatory and inhibitory neurons in which the input to each single neuron is approximately balanced. This means that the mean input current is at or slightly below threshold so that the spiking of the neuron is driven by fluctuations in its input, like in the example shown in Fig. 1.6D. Consider a simple

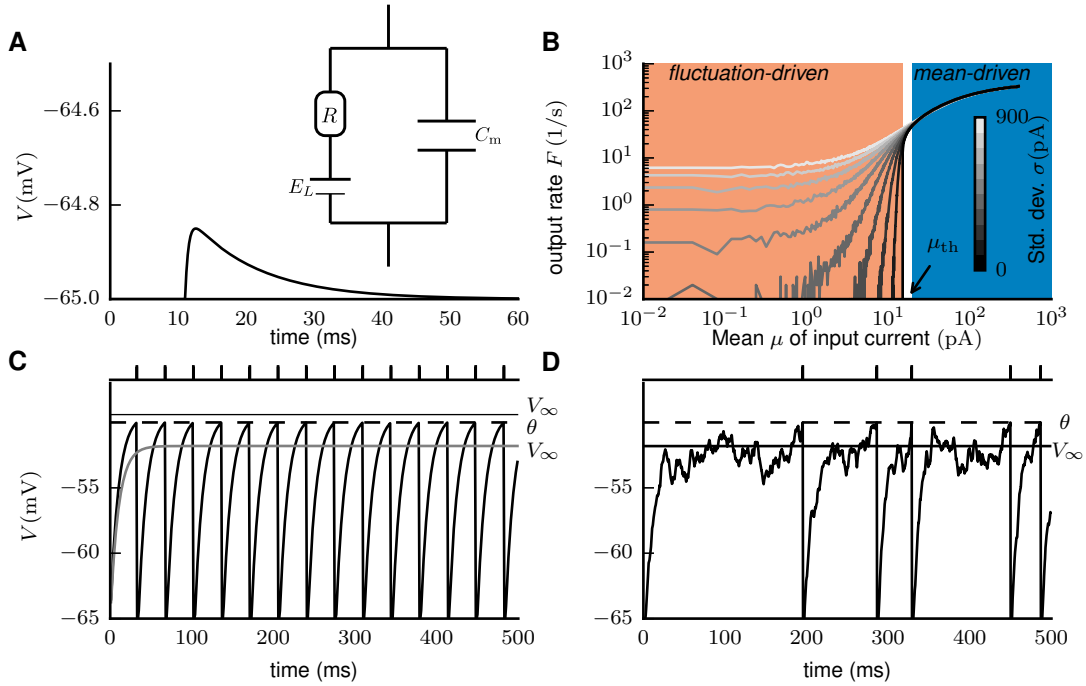

Figure 1.6: LIF neurons driven by different types of input. **A** Post-synaptic potential (PSP) of a LIF neuron in response to a single spike. The inset shows the RC circuit of the LIF neuron. **B** F-I-curve of a LIF neuron for different degrees of variance in the noisy input. **C** Responses to sub-threshold (gray, $V_\infty = -54.4\,\mathrm{mV}$) and suprathreshold (black, $V_\infty = -51.8\,\mathrm{mV}$) input. Spikes shown as vertical lines above the plot. **D** LIF neuron driven by a noise source which emits spikes drawn from a Poisson process at 7561 spikes/s such that the average free membrane potential is $V_\infty = -51.8\,\mathrm{mV}$ like the gray curve in C. The spiking of the neuron is thus caused by fluctuation in its input leading to an irregular firing pattern, displayed by vertical lines above the plot.

Name	Value	Description
τ_m	$10\,\mathrm{ms}$	membrane time constant
τ_r	$2\,\mathrm{ms}$	absolute refractory period
τ_s	$0.5\,\mathrm{ms}$	postsynaptic current time constant
C_m	$250\,\mathrm{pF}$	membrane capacity
V_r	$-65\,\mathrm{mV}$	reset potential
θ	$-50\,\mathrm{mV}$	fixed firing threshold
E_L	$-65\,\mathrm{mV}$	leak potential

Table 1.1: Single neuron parameters used throughout this work. They are identical with the parameters used in Potjans & Diesmann (2014); Schmidt et al. (2015a); Schuecker et al. (2015b).

network consisting of N_E excitatory neurons and $N_\mathrm{I} = \gamma N_\mathrm{E}$ inhibitory neurons, as specified in Table 1.2. Neurons are randomly connected to each other with a fixed probability ϵ and synaptic weights J_E for excitatory connections and $J_\mathrm{I} = -gJ_\mathrm{E}$ for inhibitory connections, i.e., the network is constructed as an Erdős-Rényi graph (Erdős & Rényi, 1959). In addition, each single cell is driven by spikes independently drawn from a Poisson process with fixed rate ν_ext and synaptic weights J_E. The weight matrix $\boldsymbol{J}$ therefore reads

$$\boldsymbol{J} = \begin{pmatrix} J_\mathrm{E} & -gJ_\mathrm{E} & J_\mathrm{E} \\ J_\mathrm{E} & -gJ_\mathrm{E} & J_\mathrm{E} \end{pmatrix}, \tag{1.2}$$

with targets in rows and source in columns. On average, a neuron receives ϵN_E excitatory and ϵN_I connections, summarized in the indegree matrix K,

$$\boldsymbol{K} = \begin{pmatrix} \epsilon N_\mathrm{E} & \epsilon N_\mathrm{I} & 1 \\ \epsilon N_\mathrm{E} & \epsilon N_\mathrm{I} & 1 \end{pmatrix}. \tag{1.3}$$

Since in this example, excitatory and inhibitory neurons receive statistically identical inputs, they emit spikes at approximately the same rate ν. A neuron thus receives on average ν spikes per second from each of its ϵN_E (ϵN_I) excitatory (inhibitory) presynaptic partners, which each lead to a total change $\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}J_\mathrm{E}$ ($-\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}J_\mathrm{E}g$) (see Sec. 6.1) in its membrane potential. Using Campbell's theorem on Poisson point processes (Papoulis & Pillai, 2002), the average input current to the neuron from within the network thus reads

$$\mu = \frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}\left(\epsilon N_\mathrm{E}J_\mathrm{E} - \epsilon\gamma N_\mathrm{E}J_\mathrm{E}g\right)\nu = \epsilon N_\mathrm{E}\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}J_\mathrm{E}\left(1-\gamma g\right)\nu \begin{cases} \leq 0 \text{ if } g \geq 1/\gamma \\ > 0 \text{ if } g < 1/\gamma \end{cases}, \tag{1.4}$$

giving a total input current

$$\mu_\mathrm{tot} = \mu + \mu_\mathrm{ext} = \mu + \frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}J_\mathrm{E}\nu_\mathrm{ext}\,. \tag{1.5}$$

Campbell's theorem also gives the variance of the input to the neuron (cf. Sec. 6.1.1),

$$\sigma^2 = \epsilon N_\mathrm{E}\left(\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}J_\mathrm{E}\right)^2\left(1+\gamma g^2\right)\nu + \left(\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\tau_\mathrm{s}J_\mathrm{E}\right)^2\nu_\mathrm{ext}. \tag{1.6}$$

To judge whether the cells in the network would be driven above threshold θ by the external input alone, we compare θ with the asymptotic free membrane potential V_∞ of a an unconnected

neuron stimulated at rate ν_{ext},

$$\eta = \frac{\nu_{\text{ext}} \frac{\tau_{\text{m}}}{C_{\text{m}}} \tau_{\text{s}} J_{\text{E}}}{\mu_{\text{th}}} = \frac{V_\infty}{\theta}\,, \tag{1.7}$$

where μ_{th} is the external drive necessary to drive an unconnected neuron to threshold. For different sets of the global parameter η, the network produces distinct dynamical states. We choose $\gamma = 0.25$ and $g = 7$, yielding an inhibition-dominated regime with $\mu < 0$ Eq. 1.4, so that increasing ν_{ext} in principle leads to a decrease of the mean membrane potential of neurons. Fig. 1.7 shows the resulting activity for different strengths of the external input. All three states considered exhibit irregular single-neuron dynamics (Fig. 1.7B).

- **Synchronous, irregular state with slow oscillations. (SI, slow)** In this regime, the external drive η is close to 1, so that the average membrane potential of the neurons is close to their threshold. Due to this proximity to the threshold, a small input fluctuation causes a large group of neurons to fire together. Inhibitory feedback subsequently leads to a shutdown of the network activity, which then recovers on the timescale of the membrane time constant $\tau_{\text{m}} = 10\,\text{ms}$, leading to slow oscillations between a silent and an active state. Spike rates are low with $\nu = 1.2$ spikes/s.

- **Asynchronous, irregular state. (AI)** Increasing the external drive to $\eta = 3$ leads to a lower average membrane potential. Consequently, small input fluctuations cause only few neurons to spike simultaneously, suppressing collective oscillations and leading to asynchronous population dynamics with an increased spike rate $\nu = 33$ spikes/s.

- **Synchronous, irregular state with fast oscillations. (SI, fast)** For an even higher external drive with $\eta = 5.5$, the network is more inhibition-dominated. A concerted increase of network activity thus leads to a strong inhibitory feedback $J < 0$ after a delay period d. As a simple assumption, the spiking rate of both populations can be described by one variable $y(t)$, since both populations statistically receive the same input, obeying the first-order delay differential equation $\frac{\text{d}}{\text{d}t} y(t) = -y(t) + J y(t-d) \overset{|J|\gg 1}{\approx} J y(t-d)$. This equation is solved by a periodic function with period $T = 4d$. A detailed derivation is given by Helias et al. (2013). Fast oscillations thus occur on the timescale of the synaptic delays at a frequency of $\sim 1/(4d)$. The average spike rate is raised to $\nu = 82.2$ spikes/s.

Similar to cortical dynamics, neurons have very low cross-correlation coefficients in the first two states, while the SI (fast) state shows increased pairwise cross-correlations. In conclusion, the simple network with only two populations and few global parameters exhibits a rich phase space with different activity regimes.

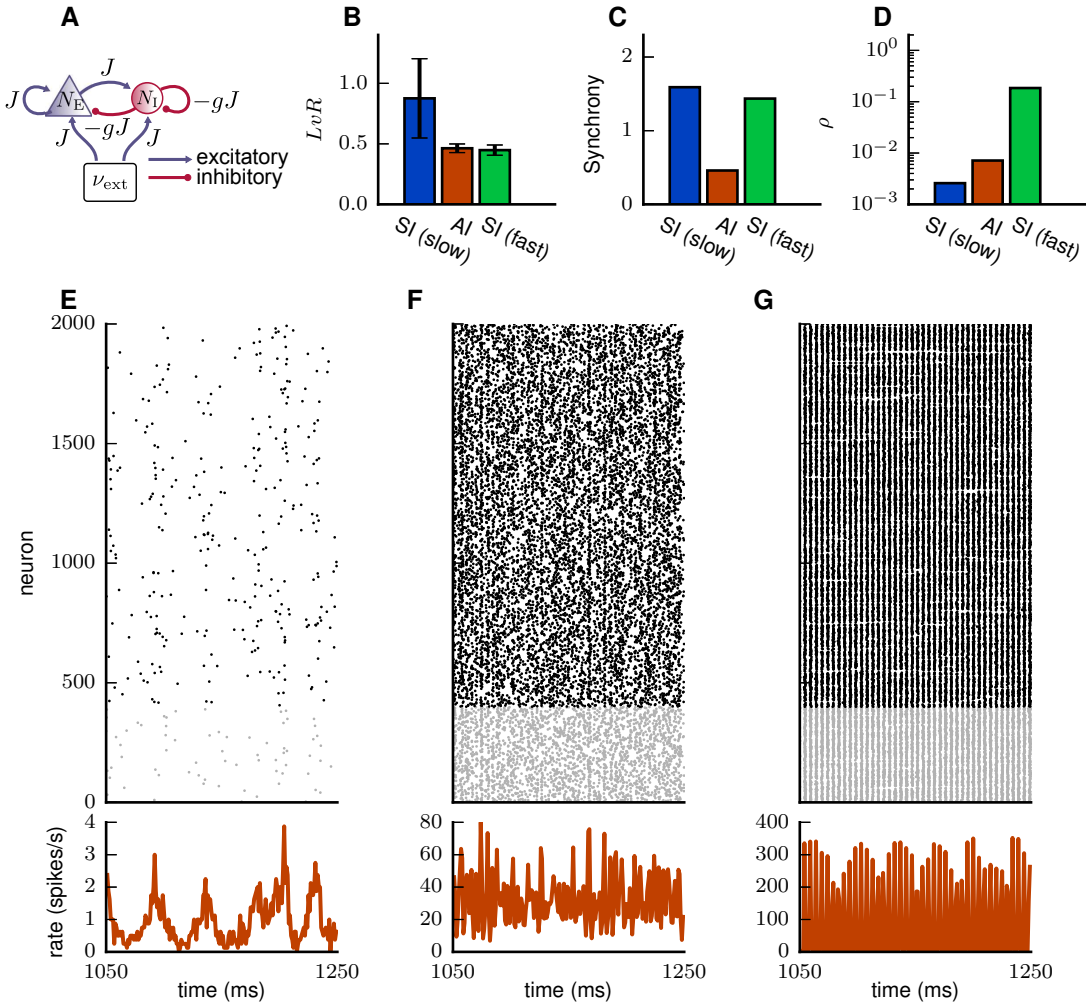

Figure 1.7: Different states of the balanced random network. **A** Sketch of the network with two populations with N_E excitatory neurons and N_I inhibitory neurons, respectively. The populations are coupled to themselves and each other and driven by Poisson input. **B** Irregularity of the single cells measured by the revised local variation of inter-spike intervals LvR for the four different states. **C** Synchrony quantified by the variance of the spike count histogram divided by its mean. **D** Average pairwise cross-correlation coefficient of the excitatory neurons. **Bottom row** Raster plots of the spiking activity for 1600 excitatory and 400 inhibitory sample neurons and the population firing rates computed on the spike trains of all excitatory neurons. **D** Synchronous, irregular regime with slow oscillations for $\eta = 0.8$. **E** Asynchronous, irregular regime for $\eta = 3$. **F** Synchronous, irregular regime with fast oscillations for $\eta = 5.5$. See Sec. 6.2 for a definition of the dynamic measures.

A: Model summary	
Populations	1 excitatory, 1 inhibitory population
Connectivity	Random connections with fixed connection probability $\epsilon = 0.1$
Neuron model	Leaky integrate-and-fire neuron, fixed voltage threshold, fixed absolute refractory period (voltage clamp)
Synapse model	Exponential-shaped postsynaptic currents
Input	Independent Poisson spike trains with fixed rate ν_{ext}
B: Populations	
Size	Excitatory population $N_{\text{E}} = 8000$, inhibitory population $N_{\text{I}} = 2000$
C: Connectivity	
Type	Random connections with independently chosen pre- and postsynaptic neurons with probability ϵ with replacement
Weights	Fixed amplitude of the post-synaptic potential of $PSP_{\max} = 0.15\,\text{mV}$, which corresponds to an amplitude of the input current of $J_{\text{E}} = 87.8\,\text{pA}$. Inhibitory weights are increased by a factor g.
Delays	Fixed delay $d = 1\,\text{ms}$
D: Neuron and synapse model	
see Table 1.1 for parameter specifications.	

Table 1.2: Description of the balanced random network model after Nordlie et al. (2009).

Insights into the network dynamics of the two-population BRN generalize to multi-population models based on the same architecture (Potjans & Diesmann, 2013). The microcircuit model of Potjans & Diesmann (2014) is one example of such a model.

1.6 The microcircuit model

The spiking network model of early sensory cortex developed by Potjans & Diesmann (2014), from now on referred to as the *microcircuit model*, forms the basis of the multi-area model of macaque visual cortex presented in Chap. 3. The model covers $1\,\text{mm}^2$ of cortical surface with 77,169 neurons connected via approximately 300 million synapses. It comprises four layers of cortex each composed of one excitatory and one inhibitory populations with proportions of $\gamma = 0.28$ (layer 2/3), 0.25 (layer 4), 0.22 (layer 5) and 0.20 (layer 6). The eight populations are connected to each other with population-specific connection probabilities derived from anatomical (Binzegger et al., 2004) and electrophysiological experiments (with the largest contribution from the review of Thomson & Lamy, 2007) on structural connectivity. Binzegger et al. (2004) used morphological reconstructions of cells in cat primary visual cortex to derive cell-type specific connection probabilities across layers and populations based on an extension of Peters's rule (cf. Sec. 1.2). Among others, Thomson & Lamy (2007) studied slices of rat cortex and tested for pairs of neurons if stimulating one cell led to a response in the other cell, indicating a direct connection between them. This yields connection probabilities in terms of

the proportions of connected pairs.

Single cells are modeled as LIF neurons with the parameters specified in Table 1.1. Importantly, inhibitory and excitatory neurons are identical except for the connections they form. The parameters are chosen such that the post-synaptic potentials match the in-vivo situation, with reduced membrane resistance due to a combination of synaptic input and spikes (Destexhe et al., 2003; Waters & Helmchen, 2006).

The circuit reproduces important aspects of cortical dynamics. Neurons spike irregularly with low synchrony across cells and heterogeneous firing rates across populations (Fig. 1.8). Inhibitory neurons exhibit higher rates than excitatory cells despite their identical parameters, showing that the complex connectivity of the circuit itself can produce this feature of cortical dynamics. All populations exhibit fast oscillations in the range of ~ 80 Hz. Another result of the study is a hypothesis for the propagation of transient thalamic input, which is testable in an experiment. The input enters the circuit preferentially in layers 4 and 6 and then propagates throughout the network (Fig. 1.8C). Stable propagation of activity is achieved via a "handshaking" mechanism. For instance, layer 4 excites layer 2/3, which then activates the inhibitory population of layer 4, leading to a relaxation of layer 4 back to its unperturbed ground state. The E-I-specificity of the inter-laminar projections, i.e., the relative proportion of synapses targeting excitatory and inhibitory neurons, plays a crucial role for this by preventing a disproportionate increase of excitatory activity following the additional, external input.

In conclusion, the microcircuit model combines simple single-neuron dynamics with complex connectivity and demonstrates the strong impact of structural connectivity on the network dynamics. However, due to its limited size, it fails to capture important aspects of cortical dynamics such as long-range interactions and the dominant contribution of low frequencies to the power spectrum. Furthermore, the model is under-constrained because only half of the synapses received by each neuron is actually modeled while the other half originates outside of the $1\,\mathrm{mm}^2$ patch and are thus replaced by stochastic input. Extending the model to multiple areas as described in Chap. 3 therefore is a logical next step.

1.7 Mean-field theory of balanced random networks

The stationary state of a balanced random network can be computed using mean-field theory. In the case of high indegrees K (incoming synapses per neuron) and low synaptic weights J_E, the dynamics of LIF neurons can be simplified using the *diffusion approximation* (Fourcaud & Brunel, 2002). The input spike trains to the neurons are approximated by a current fluctuating randomly around the mean μ with variance σ^2. The fluctuations are drawn from a Gaussian

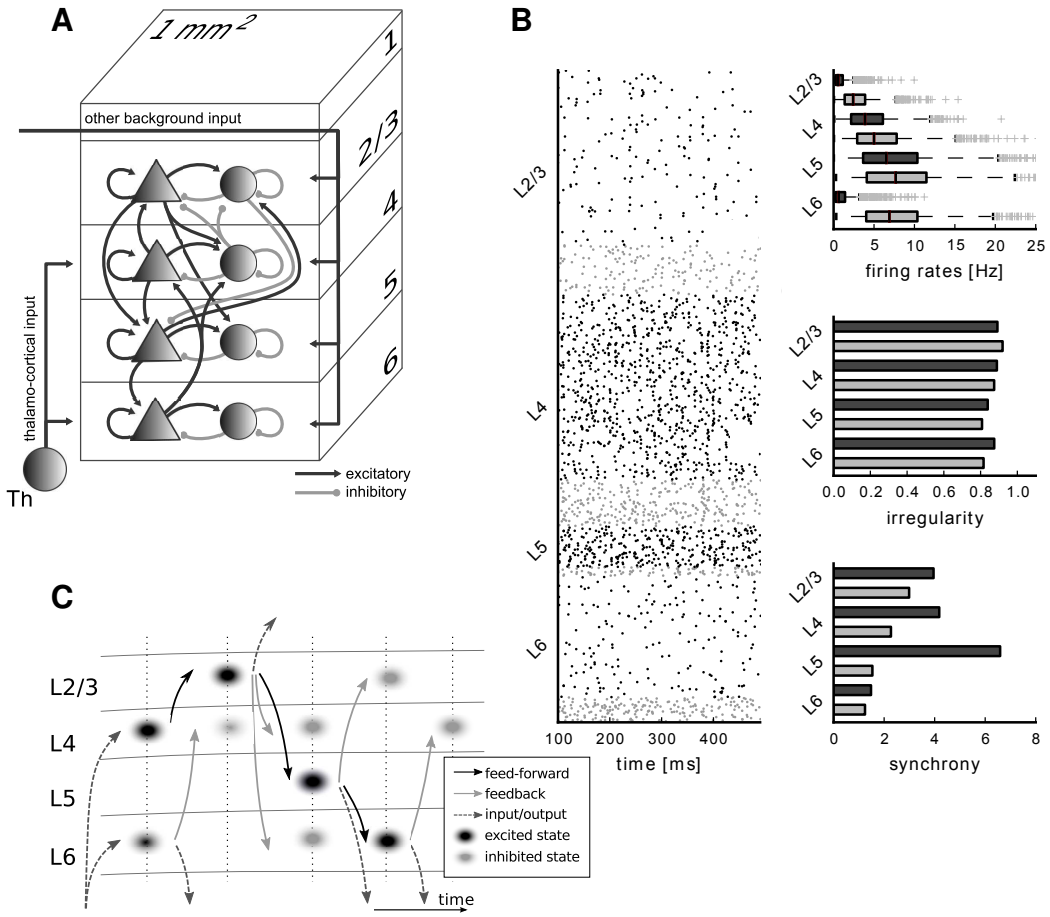

Figure 1.8: The microcircuit model of early sensory cortex. Figures adapted with permission from Potjans & Diesmann (2014). **A** Sketch of the microcircuit model with four layers and one excitatory (triangles) and one inhibitory (circles) population per layer. Arrows indicate connections between populations. Each population is driven by a constant background noise provided by spike trains drawn from a Poisson process. **B** Stationary state of the model. Left: Raster plot of the spiking activity with excitatory populations in black and inhibitory populations in gray. Right: Box plot (Tukey, 1977) of the population-averaged firing rates (top), irregularity measured by the coefficient of variation of the inter-spike interval distribution (CV ISI) (middle) and synchrony quantified by the variance of the spike count histogram divided by its mean (bottom). **C** Sketch of the propagation of transient thalamic input in the circuit.

white noise process $\xi(t)$ with mean $\langle\xi(t)\rangle = 0$ and $\langle\xi(t)\xi(t'\rangle = \delta(t-t')$, so that Eq. 1.1 becomes

$$\begin{aligned} \tau_{\mathrm{m}}\frac{\mathrm{d}V}{\mathrm{d}t} &= -V + I_{\mathrm{s}}(t) \\ \tau_{\mathrm{s}}\frac{\mathrm{d}I_{\mathrm{s}}}{\mathrm{d}t} &= -I_{\mathrm{s}} + \mu + \sigma\sqrt{\tau_{\mathrm{m}}}\xi(t)\,, \end{aligned} \tag{1.8}$$

where we set $E_{\mathrm{L}} = 0$ without compromising generality and the membrane capacitance has been absorbed into the definition of the synaptic current. In the case of homogeneous neuronal populations considered here, the stationary activity of a neuronal population i is well described by a population-averaged firing rate ν_i caused by input current μ_i, σ_i. Using the Fokker-Planck formalism, the population firing rate can then be shown to follow the Siegert formula (Siegert, 1951; Fourcaud & Brunel, 2002),

$$\begin{aligned} \frac{1}{\nu_i} &= \tau_{\mathrm{r}} + \tau_{\mathrm{m}}\sqrt{\pi}\int_{\frac{V_r-\mu_i}{\sigma_i}+\gamma\sqrt{\frac{\tau_{\mathrm{s}}}{\tau_{\mathrm{m}}}}}^{\frac{\theta-\mu_i}{\sigma_i}+\gamma\sqrt{\frac{\tau_{\mathrm{s}}}{\tau_{\mathrm{m}}}}} e^{x^2}\,(1+\mathrm{erf}(x))\,\mathrm{d}x \\ &=: 1/\Phi_i(\boldsymbol{\nu}) \end{aligned} \tag{1.9}$$

$$\mu_i = \tau_{\mathrm{m}}\sum_j K_{ij}J_{ij}\nu_j + \tau_{\mathrm{m}}K_{\mathrm{ext}}J_{\mathrm{E}}\nu_{\mathrm{ext}} \tag{1.10}$$

$$\sigma_i^2 = \tau_{\mathrm{m}}\sum_j K_{ij}J_{ij}^2\nu_j + \tau_{\mathrm{m}}K_{\mathrm{ext}}J_{\mathrm{E}}^2\nu_{\mathrm{ext}} \tag{1.11}$$

which holds up to linear order in $\sqrt{\tau_{\mathrm{s}}/\tau_{\mathrm{m}}}$ and where $\gamma = |\zeta(1/2)|/\sqrt{2}$, with ζ denoting the Riemann zeta function (Abramowitz & Stegun, 1974). A detailed derivation is given in Sec. 6.1.2. Eq. 1.9 is a self-consistent equation which has to be solved numerically and yields the population-averaged firing rates of a multi-population network model. It forms the basis of the theoretical method described in Chap. 4.

1.8 Network modeling studies

Most studies on bottom-up models of cortex have focused either on simulating local circuits with varying degrees of detail or on modeling the whole cortex while representing each area with highly simplified models. This section reviews a selection of these studies and their findings, starting with the small-scale models.

Traub et al. (2005) constructed a model of a single cortical column coupled to a network mimicking thalamic structures, similar to the microcircuit model, with 3,560 multi-compartment neurons including a large range of different cell types with diverse dynamical properties. The model reproduces experimental findings such as oscillations on the order of 30 Hz as well as faster population dynamics including sudden bursts of activity. Haeusler et al. (2009) compared the two experimentally compiled connectivity maps of Binzegger et al. (2004) and Thomson et al.

(2002), which were also used to construct the microcircuit model of Potjans & Diesmann (2014), by testing the computational performance of spiking network models based on each data set. Both models accomplish several tasks equally well and the distribution of degrees in the network is crucial for its performance. Hill & Tononi (2005) studied the emergence of slow oscillations in a model of two cortical areas coupled to thalamus, in relation to the transition from wakefulness to sleep. They found that cortico-cortical connections between the areas synchronize the slow oscillations appearing in the sleep state. Markram et al. (2015) presented a model of a cortical column of rat somato-sensory cortex which integrates a tremendous amount of details of the biological tissue including cell morphologies and cell and synapse dynamics, based on a large body of experimental studies (see also Fig. 1.2). Neurons are represented by multi-compartment models including detailed dynamics of ion channels and short-term synaptic plasticity. Their reference column comprises 31,000 neurons and 36 million synapses. They combined seven instantiations of it to a *mesocircuit* and reproduced experimental observations on spontaneous activity, propagation of activity induced by transient stimulation and the suppression of spike correlation by inhibitory feedback in the network.

Other studies investigate large-scale dynamics by including many cortical areas, represented by simple models. Deco et al. (2009) used a connectivity matrix based on data from the CoCoMac database (Stephan et al., 2001) to couple Wilson-Cowan oscillators, representing the average firing rate in cortical areas. They found two groups of areas anti-correlated to each other on a very slow time scale (~ 0.1 Hz) and a high sensitivity of the system to variations of the system parameters, i.e., delays, coupling strength and the level of external noise. Cabral et al. (2011) used the Kuramoto model to represent single areas, an oscillator model mimicking an E-I-circuit in an oscillatory state so that the network state can be described by the oscillation phase only. They connected the areas according to human DTI data from Hagmann et al. (2008) and found that the large-scale network behavior is robust against changes in the single node dynamics. The model best reproduces empirical data on functional connectivity for large delays where synchronized clusters of areas form which mainly correspond to clusters in the structural connectivity. Deco & Jirsa (2012) used the same DTI data to connect single areas modeled as fully connected spiking networks with 100 excitatory and 100 inhibitory neurons receiving stochastic background input. They varied the coupling strength between areas and observed that experimental functional connectivity is best reproduced when the model is close to a transition from a single-attractor to a multi-attractor phase space. From this, they concluded that the brain is in a critical state where it explores the attractors which have not emerged yet, but appear in the phase space either as saddle points with at least one instable direction or as regions with slow flow. Izhikevich & Edelman (2008) constructed a model comprising 1 million multi-compartmental spiking neurons connected via $\sim 0.5 \cdot 10^9$ synapses, corresponding to an

average of 500 incoming synapses per neuron. The model featured 22 different neuron types and a high degree of detail in the synapses. It produces self-sustained, chaotic activity and multiple functional clusters. However, such downscaling of synaptic indegrees generally affects network activity already in terms of first- and second-order statistics and leads to quantitative and sometimes even qualitative deviations with respect to a full-scale model (van Albada et al., 2015).

While these models have elucidated various aspects of cortical dynamics, how local dynamics inside areas shapes cortical interactions on a global scale remains a largely open question.

1.9 The NEST simulator

Several software simulators for the simulation of spiking neural networks are available, two prominent examples being NEST (NEural Simulation Tool) (Gewaltig & Diesmann, 2007) and NEURON (Carnevale & Hines, 2006). While NEURON is specialized in the simulation of detailed, multi-compartment neurons, NEST focuses on the efficient simulation of networks of point neurons. NEST is an open-source software project developed by the NEST Initiative (www.nest-initiative.org). Its development started in 1994 and is nowadays supported by a team of around 20 software developers. The software currently includes more than 20 neuron models and 9 models of synapses including various forms of synaptic plasticity such as spike-timing dependent plasticity. A single neuron is represented by a system of few differential equations with pointwise non-linearity and neurons interact with each other via spikes. From a computational point of view, the software thus has to deal with simple differential equations coupled via delayed, pulsed δ-type interactions. Depending on the neuron type, the differential equations can be solved either exactly (Rotter & Diesmann, 1999; Plesser & Diesmann, 2009) or with a numerical solver. Due to the relatively simple nature of these equations, the computational challenge lies in the efficient communication of the spikes across the network, which is performed in steps of the minimal delay in the network (Morrison et al., 2005). NEST runs on systems such as laptops with one to few cores, middle-sized compute clusters ($\sim 10^2 - 10^3$ compute cores), and the largest modern supercomputers with up to 10^6 compute cores. This flexibility has implications for the design of the memory structures for neurons and synapses and rules out solutions dedicated to a specific type of hardware. Important steps in the past development of the software include:

- Introduction of parallel computing using the message-passing interface (MPI) enabling the simulation of a network of 10^5 neurons with 10^9 synapses, representing a realistic indegree per neuron (Morrison et al., 2005).

- Introduction of the hybrid simulation scheme combining MPI and OpenMP multi-threading (Plesser et al., 2007).
- Development of the third-generation NEST kernel (NEST 3g) with scalability up to 10^4 cores (Helias et al., 2012).

In NEST, neurons are equally distributed across threads, also termed *virtual processes* (VP). Fig. 2.3A shows the infrastructure of the 3g kernel. Each VP holds a sparse table with single-bit entries for all neurons of the network. If a neuron has a connection to at least one neuron locally stored on the VP, the sparse table holds a pointer to a *target list*, a vector with a structure for each synapse type (e.g. a static or a plastic synapse), which stores the synapse parameters including the local targets. In the case of large machine sizes and sparsely connected networks, a neuron has its targets distributed across many virtual processes and thus one to few targets at most on a given VP, making this memory infrastructure disadvantageous due to the required memory overhead. We therefore developed the fourth-generation simulation kernel (NEST 4g), described in Chap. 2. It introduces a new infrastructure for the efficient storage of the connections, achieving good scalability up to 10^5 cores. This development was released as NEST 2.6. The new kernel enabled a world-record simulation of $1.86 \cdot 10^9$ neurons connected via a total of $11.1 \cdot 10^{12}$ synapses (RIKEN BSI, 2013).

II

Results

CHAPTER 2

Spiking network simulation code for petascale computers

The following chapter has been published in Kunkel S, Schmidt M, Eppler JM, Plesser HE, Masumoto G, Igarashi J, Ishii S, Fukai T, Morrison A, Diesmann M and Helias M (2014) Spiking network simulation code for petascale computers. *Front. Neuroinform. 8:78* (Kunkel et al., 2014).

Author contributions: The author's contribution to this study was help with the implementation of the new synaptic infrastructure (described in Sec. 2.3.2 and Sec. 2.3.3), design, execution, and interpretation of the performance measuring benchmarks described in Sec. 2.2.5, Sec. 2.3.6, and contributions to the writing of the manuscript.

2.1 Introduction

In a previous study (Kunkel et al., 2012b), we presented data structures that allow the neuronal network simulator NEST (Gewaltig & Diesmann, 2007) to exploit the increasingly available supercomputers such as JUQUEEN and the K computer (Helias et al., 2012). Although we could carry out benchmarks utilizing over 100,000 cores, analysis of the memory consumption (Sec. 2.3.1) reveals that at such large sizes, the infrastructure required on each machine to store synapses with local targets becomes the dominant component.

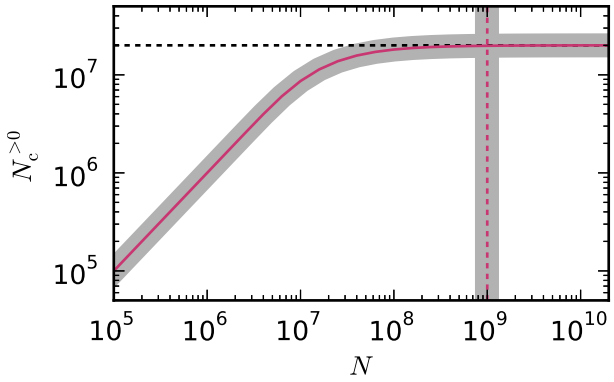

Figure 2.1: Number of local target lists approaches the number of local synapses. The gray curve shows the expected number of target lists per core $N_c^{>0}$, given by Eq. 2.6, that contain at least one synapse as a function of the total network size N. The pink curve $(1 - \exp(-K_{\mathrm{VP}}/N))\,N$ is an approximation (Eq. 2.7) for large networks. Here, each core represents $N_{\mathrm{VP}} = 2000$ neurons with $K = 10,000$ synapses per neuron, which is a realistic example. In the limit of large N the number of target lists approaches the number of local synapses $K_{\mathrm{VP}} = KN_{\mathrm{VP}}$ (black dashed horizontal line). The gray vertical line marks the network size N_ζ, given by Eq. 2.9, at which the number of target lists has reached $\zeta = 99\%$ of the limit K_{VP}. The pink dashed vertical line at $K_{\mathrm{VP}}/2/(1-\zeta)$ is an approximation (Eq. 2.10) of the full expression Eq. 2.9.

The reason can be understood with a simple calculation. Assuming each neuron contacts 10,000 other neurons, and that these neurons are randomly distributed over the entire supercomputer, then on a system with 100,000 cores, the probability of a core containing more than one target neuron is rather small, and the majority of cores will contain no target neurons of the given neuron. This is illustrated in Fig. 2.1; as the total network size increases (whilst maintaining a constant number of neurons on a core), the number of target lists of non-zero length approaches the number of synapses on the local core and, consequently, each target list has an expected length of 1. As each new target list comes with a certain memory overhead, the average total costs per synapse increase with increasing network size. This acceleration in memory consumption only stops when each target list is either of length 0 or length 1 and from then on each synapse carries the full overhead of one target list. With realistic parameters, the largest networks that can be represented on supercomputers using the technology employed in

Helias et al. (2012) reach this limit.

The memory model introduced in Kunkel et al. (2012b) triggered the major advance of using sparse tables to store incoming connections, thus reducing the memory overhead for target lists of length 0. However, the memory overhead for neurons with local targets is still substantial, as the data structure is designed to enable variable amounts of heterogeneous connection types, e.g. static synapses and various types of plasticity including short-term plasticity (Tsodyks et al., 1998, 2000) and spike-timing dependent plasticity (see e.g. Morrison et al., 2008, for a review). Nevertheless, as the network size increases, it becomes increasingly common that a neuron has only one local target, thus the majority of this structure is redundant: both the number (just 1) and the type of the connection are known.

Thus the challenge is to develop a data structure that is streamlined for the most common case (on large systems) that a neuron has one local target, and yet allows the full flexibility with respect to connection type and number when needed. A constraint on the granularity of the parallelization is the assumption that the neuron objects themselves are not distributed but viewed as atomistic and simulated on a single compute node as an entity. Consequently the connection infrastructure only needs to represent synaptic connections, not connections between different compartments of a neuron. In Sec. 2.3.2, we present a data structure that fulfills these criteria by self-collapsing along two dimensions: the heterogeneity collapses to a single well-defined connection type and the dynamic length of the connections vector collapses to a single element. Moreover, a redesign of the synapse data structure and the handshaking algorithm at connection time (see Sec. 2.3.3) allow the polymorphism of the synapse types to be exploited without requiring a pointer to a virtual function table, thus saving 8 B for every synapse in the network. Making use of the limited number of neurons local to a core further allows us to replace the full 8 B target pointer by a 2 B index and combining the delay and the synapse type into a joint data structure saves another 8 B. The creation of many small synapse objects presents a challenge to the memory allocation which we meet by implementing a dedicated pool allocator in Sec. 2.3.4. Finally, in Sec. 2.3.5 a new data structure to store local neurons is introduced. The sparse node array exploits the regularity in assigning neurons to machines and thereby eliminates memory overhead for non-local neurons at the expense of an increased number of search steps to locate a node. The major result of this study is that we are now capable of employing the full size of currently available petascale computers such as JUQUEEN and K.

Sec. 2.2.1 specifies the neuronal network model used to obtain quantitative data. Sec. 2.2.2 extends the mathematical memory model previously introduced in Kunkel et al. (2012b) which we use to analyze the scaling properties of alternative data structures. In Sec. 2.3.1 we first investigate the memory consumption of NEST on petascale computers in detail. In the following

sections we describe the data structures, corresponding algorithms, and the allocator outlined above. Finally in Sec. 2.3.6 we quantitatively compare the resulting new (4g) simulation code with the previous (3g) version on the two supercomputers JUQUEEN and K. The capability of the fourth generation code is demonstrated by orchestrating all the available main memory of the K computer in a single simulation. In the concluding section we assess our achievements in terms of flexibility and network size in the light of previous work and discuss limitations.

This work concludes a co-development project for the K computer in Kobe, which started in 2008 (Diesmann, 2013). Preliminary results have been published in abstract form (Diesmann, 2012; Kunkel et al., 2013) and as a joint press release of the Jülich Research Centre and RIKEN (RIKEN BSI, 2013). The conceptual and algorithmic work described here is a module in our long-term collaborative project to provide the technology for neural systems simulations (Gewaltig & Diesmann, 2007).

2.2 Methods

2.2.1 Network model

All measurements of memory usage and run time are carried out for a balanced random network model (Brunel, 2000) of 80% excitatory and 20% inhibitory integrate-and-fire neurons with alpha-shaped post-synaptic currents. Both types of neurons are represented by the NEST model `iaf_neuron` with a homogeneous set of parameters. All excitatory-excitatory connections exhibit spike-timing dependent plasticity and all other connections are static. Simulations performed with the 3rd generation simulation kernel (Kunkel et al., 2012b; Helias et al., 2012) employ the models `stdp_pl_synapse_hom` and `static_synapse` whereas simulations run with the 4th generation simulation kernel presented in this manuscript use the novel high-performance computing (HPC) versions of these models (`stdp_pl_synapse_hom_hpc` and `static_synapse_hpc`, described in Sec. 2.3.3). We use two sets of parameters for the benchmarks. Within each set, the only parameter varied is the network size in terms of number of neurons N.

Set 1 The total number of incoming connections per neuron is fixed at $K = 11,250$ (9000 excitatory, 2250 inhibitory). The initial membrane potentials are drawn from a normal distribution with $\mu = 9.5\,\mathrm{mV}$ and $\sigma = 5.0\,\mathrm{mV}$. The initial synaptic weights are set to $J_\mathrm{E} = 45.61$ pA for excitatory and to $J_\mathrm{I} = -gJ_\mathrm{E}$, $g = 5$ for inhibitory synapses. All neurons receive excitatory external Poissonian input causing a mean membrane potential of $\eta\,\theta = \tau_\mathrm{s} J_\mathrm{E} \frac{\tau_\mathrm{m}}{C_\mathrm{m}} \nu_\mathrm{ext}$. With $\eta = 1.685$, $\theta = 20\,\mathrm{mV}$, $\tau_\mathrm{m} = 10\,\mathrm{ms}$, $C_\mathrm{m} = 250$ pF and $\tau_\mathrm{s} = 0.3258\,\mathrm{ms}$, this corresponds to the input spike rate of $\nu_\mathrm{ext} = \eta \frac{\theta}{\tau_\mathrm{s} J_\mathrm{E} \frac{\tau_\mathrm{m}}{C_\mathrm{m}}} \simeq 20,856$

spikes per second summed over all external inputs to a neuron. Within the simulation period of 1 s, each neuron fires on average 7.6 times. Spikes are communicated every 1.5 ms, corresponding to the synaptic delay, but neuronal state variables are advanced in steps of 0.1 ms. For further details of the network model such as neuronal and synaptic parameters please see the example script `hpc_benchmark.sli`, which is available in the next major release of NEST.

Set 2 For the second set of benchmarks, the number of incoming connections per neurons is reduced to $K = 6000$. The other parameters are adapted to obtain an irregular activity state with an average rate of 4.5 spikes per second. The adapted parameters are $J_{\mathrm{E}} = 50\,\mathrm{pA}$, $g = 7$ and $\eta = 1.2$. All other parameters are the same as in set 1.

2.2.2 Memory-usage model

Our efforts to redesign the objects and fundamental data structures of NEST towards ever more scalable memory usage are guided by the method introduced in Kunkel et al. (2012b). The method is based on a model which describes the memory usage of a neuronal network simulator as a function of the network parameters, i.e., the number of neurons N and the number K of synapses per neuron as well as the parameters characterizing the distribution of the simulation code over the machine, i.e., the number of compute nodes M and the number of threads T running on each compute node. Threads are also termed "virtual processes" due to NEST's internal treatment of threads as if they were MPI processes by completely separating their memory areas. This replication of data structures for each thread is reflected in expressions of the memory consumption of the synaptic data structures that depend only on the product MT, as shown in Sec. 2.2.3. In the following, we therefore use the terms "the total number of virtual processes" synonymously with "the total number of threads", both referring to MT. We apply the model to NEST to determine the data structures that dominate the total memory usage at a particular target regime of number of virtual processes. Once the critical parts of NEST have been identified, the model enables us to predict the effect of potential design changes for the entire range of the total number of threads from laptops to supercomputers. Furthermore, the model assists the benchmarking process as it facilitates the estimation of the maximum network size that just fits on a given number of compute nodes using a given number of threads each. We briefly restate the model here and describe the required alterations to the model for the petascale regime, which allow a more precise assessment of the contributions of different parts of infrastructure. For further details on the model and its practical application, please refer to our previous publications (Kunkel et al., 2012b,a; Helias et al., 2012).

Three main components contribute to the total memory consumption of a neuronal net-

work simulator: the base memory usage of the simulator including external libraries such as MPI, $\mathcal{M}_0(M)$, the additional memory usage that accrues when neurons are created, $\mathcal{M}_\mathrm{n}(M,N)$, and the additional memory usage that accrues when neurons are connected, $\mathcal{M}_\mathrm{c}(M,T,N,K)$. The memory consumption per MPI process is given by

$$\begin{aligned}\mathcal{M}(M,T,N,K) &= \mathcal{M}_0(M) + \mathcal{M}_\mathrm{n}(M,N) \\ &\quad + \mathcal{M}_\mathrm{c}(M,T,N,K)\,.\end{aligned} \tag{2.1}$$

As suggested in Kunkel et al. (2012b), we determined $\mathcal{M}_0(M)$ by measuring the memory usage of NEST right after start-up, which was at most 268 MB on the K computer and 26 MB on JUQUEEN. However, in this study $\mathcal{M}_0(M)$ also accounts for the communication buffer that each MPI process requires in order to receive spike information from other processes. As NEST uses `MPI_Allgather` to communicate spike data, the buffer grows proportionally with the number of MPI processes M. Hence, in the petascale regime the contribution of this buffer to the total memory usage is no longer negligible. Here, we assume that each MPI process maintains an outgoing buffer of size 1000, where each entry consumes 4B, such that the memory that is taken up by the incoming buffer amounts to $M \times 4\,\mathrm{kB}$. In NEST the communication buffers increase dynamically whenever the instantaneous rate of the simulated network requires more spikes to be communicated. In simulations of the benchmark network model described in Sec. 2.2.1 we measured send-buffer sizes of 568 entries (in a full K computer simulation), such that for this model the assumed buffer size of 1000 is a worst-case scenario.

Neuron objects in NEST are distributed across virtual processes in a round-robin fashion and connections are represented on the process of their post-synaptic neuron. We use the term "VP-local" to indicate that a neuron is local to a certain virtual process. As neurons with similar properties are typically created en bloc, the round-robin distribution scheme constitutes a simple form of static load-balancing for heterogeneous networks with varying numbers of incoming connections per neuron. If each virtual process owns sufficiently many neurons, the number of local connection objects is similar across processes. Therefore, in our model we let $N_M = N/M$ and $K_M = N_M K$ denote the average number of neuron and connection objects per MPI process, and we let $N_\mathrm{VP} = N_M/T$ and $K_\mathrm{VP} = N_\mathrm{VP}K$ denote the average number of VP-local neuron and connection objects.

In the regime of $\sim 10,000$ virtual processes, for a randomly connected network the targets of a neuron become more and more spread out. This results in the limiting case where K processes each own one of the targets and the remaining $MT - K$ processes do not own any of the targets. As a consequence, the connection infrastructure becomes increasingly sparse,

where the extent of sparseness can be quantified in a probabilistic way. Here we use the symbol $\sim$ reading "on the order of" (Hardy & Wright, 1975, page 7) in the physics sense (Jeffreys & Jeffreys, 1956, page 23). This relation stating that two quantities are not differing by more than a factor of 10 needs to be distinguished from the big-O notation below (Sec. 2.2.3), which is used to describe the limit of a function.

To quantify the sparseness we define $p_\emptyset$ and p_1 as the probabilities that a particular neuron has 0 or 1 local target on a given virtual process, respectively. Each neuron draws on average K source neurons from the set of N possible source neurons. If the incoming connections per neuron are drawn independently, on average K_{VP} source neurons are drawn on each virtual process. Due to the large numbers, the distribution around this mean value is narrow. The probability that a particular neuron is drawn as a source is $1/N$ and the probability that the neuron is not drawn as a source is $1-1/N$. We can therefore adopt the simplifying assumption that $p_\emptyset = (1-1/N)^{K_{\mathrm{VP}}}$ expresses the average probability that a neuron does not connect to any VP-local target, such that $N_{\mathrm{c}}^{\emptyset} = p_\emptyset N$ denotes the expected number of neurons without any VP-local target. In this study we adapt the model to separately account for the neurons with exactly one VP-local target and for those with more than one VP-local target. We introduce $p_1 = (1-1/N)^{K_{\mathrm{VP}}-1} K_{\mathrm{VP}}/N$ as the average probability that a neuron has exactly one local target, such that $N_{\mathrm{c}}^{1} = p_1 N$ denotes the expected number of neurons with only one local target. The remaining $N - N_{\mathrm{c}}^{\emptyset} - N_{\mathrm{c}}^{1}$ neurons connect to more than one VP-local target.

Throughout this study, we keep the average number of incoming connections per neuron fixed at either $K = 11,250$ or $K = 6000$ in accordance with the two employed benchmark network models (see Sec. 2.2.1), and we assume $T = 8$ threads per MPI process, which corresponds to the maximum number of threads per node supported on the K computer (see Sec. 2.2.4). We explicitly differentiate between connections with spike-timing dependent plasticity and connections with static weights. This is a trivial but useful extension to the model, which enables a more precise prediction of memory usage. For the case that all excitatory-excitatory connections exhibit STDP, the number of STDP connections per MPI process amounts to $K_M^{\mathrm{stdp}} = K_M \beta^2$, where $\beta = 0.8$ is the fraction of excitatory neurons, and the remaining $K_M^{\mathrm{stat}} = K_M - K_M^{\mathrm{stdp}}$ synapses are static. In Helias et al. (2012), this differentiation between two connection types was not required as in NEST 2.2 (3g kernel) the employed models `stdp_pl_synapse_hom` and `static_synapse` have an identical memory usage of $m_{\mathrm{c}}^{\mathrm{stdp}} = m_{\mathrm{c}}^{\mathrm{stat}} = 48\,\mathrm{B}$.

With the above definitions, the memory consumption of the latter two terms of Eq. 2.1 can be further decomposed into

	parameter	description	value in B	
			3g	4g
$\mathcal{M}_\mathrm{n}(M,N)$	m_n	memory usage of one neuron object of type `iaf_psc_alpha`	1100	
	m_n^0	memory overhead per neuron	0.33	0
	m_n^+	memory overhead per local neuron	16	24
	$m_\mathrm{n}^\emptyset$	memory overhead per non-local neuron	0	
$\mathcal{M}_\mathrm{c}(M,T,N,K)$	$m_\mathrm{c}^\mathrm{stat}$	memory usage of one connection object of type `static_synapse` (3g) or `static_synapse_hpc` (4g)	48	16
	$m_\mathrm{c}^\mathrm{stdp}$	memory usage of one connection object of type `stdp_pl_synapse` (3g) or `stdp_pl_synapse_hpc` (4g)	48	24
	m_c^0	memory overhead per neuron	0.33	
	m_c^1	memory overhead per neuron with one local target	96	24
	$m_\mathrm{c}^{>1}$	memory overhead per neuron with more than one local target	160	128
	$m_\mathrm{c}^\emptyset$	memory overhead per neuron without local targets	0	

Table 2.1: Parameter definitions and values of memory-usage model for 3g and 4g technology. The top part of the table summarizes the parameters relevant for the memory consumption due to the neuronal infrastructure $\mathcal{M}_\mathrm{n}(M,N)$, the bottom part shows the parameters determining the memory consumption $\mathcal{M}_\mathrm{c}(M,T,N,K)$ of the synaptic infrastructure. Lower case symbols m refer to the memory per object, where objects are neurons or connections, respectively. The columns "3g" and "4g" distinguish between the parameter values for the two kernel versions.

$$\begin{aligned}\mathcal{M}_\mathrm{n}(M,N) &= Nm_\mathrm{n}^0 + (N-N_M)\,m_\mathrm{n}^\emptyset \\ &\quad + N_M\left(m_\mathrm{n}^+ + m_\mathrm{n}\right) \end{aligned} \tag{2.2}$$

$$\begin{aligned}\mathcal{M}_\mathrm{c}(M,T,N,K) &= TNm_\mathrm{c}^0 + TN_\mathrm{c}^\emptyset m_\mathrm{c}^\emptyset \\ &\quad + TN_\mathrm{c}^1 m_\mathrm{c}^1 + T(N-N_\mathrm{c}^\emptyset - N_\mathrm{c}^1)m_\mathrm{c}^{>1} \\ &\quad + K_M^\mathrm{stat} m_\mathrm{c}^\mathrm{stat} + K_M^\mathrm{stdp} m_\mathrm{c}^\mathrm{stdp}\end{aligned} \tag{2.3}$$

in order to capture the contributions of neuron and synapse objects and different parts of infrastructure. Table 2.1 summarizes the model parameters required to specify $\mathcal{M}_\mathrm{n}(M,N)$ and $\mathcal{M}_\mathrm{c}(M,T,N,K)$ and contrasts their values for the 3g (Helias et al., 2012) and 4g simulation technology. For convenience, we provide the values already at this point even though they are explained only in Sec. 2.3.

Note that we assume the same overhead $m_c^{>1}$ for all neurons with more than one local target, which means that we do not introduce any further distinction of possible synapse containers for the cases where more than one synapse needs to be stored (see Sec. 2.3.2 for the details). Here, we set $m_c^{>1}$ such that it corresponds to the most complex synapse container that can occur in simulations of the benchmark network model described in Sec. 2.2.1, which is a container that stores two different types of synapses in corresponding `vectors`. As a result of this worst-case assumption the model produces a slight overestimation of memory consumption.

Overall, however, the model underestimates the effectively required memory resources as the theoretically determined parameter values that we employ here reflect only the memory usage of the involved data types on a 64 bit architecture, but they cannot account for the memory allocation strategies of the involved dynamical data structures (Kunkel et al., 2012b).

2.2.3 Number and length of local target lists

Using the notation of Sec. 2.2.2 the probability of a neuron to be the source of a particular synapse is $1/N$ and consequently the probability of not being the source of any of the $K_{\mathrm{VP}} = KN/(MT)$ VP-local synapses is

$$p_\emptyset = \left(1 - \frac{1}{N}\right)^{\frac{KN}{MT}}. \tag{2.4}$$

Empty target lists are not instantiated and therefore do not cause overhead by themselves. We recognize in Eq. 2.4 the structure $(1 + x/N)^N$ exposed by

$$p_\emptyset \quad = \quad \left[\left(1 - \frac{1}{N}\right)^{N}\right]^{\frac{K}{MT}}.$$

Thus in the limit of large N we can use the definition of the exponential function $\lim_{N\to\infty}(1 + x/N)^N = \exp(x)$ to replace the term $[\cdot]$. Conceptually this corresponds to the approximation of the binomial probabilities $\begin{pmatrix} N \\ k \end{pmatrix} p^k(1-p)^{N-k}$ by the corresponding Poisson probabilities $\frac{\lambda^k}{k!}\exp(-\lambda)$, where $\lambda = Np = \text{const.}$ as $N \to \infty$. In this limit we have

$$\begin{aligned} \tilde{p}_\emptyset &= e^{-\frac{K}{MT}} \\ &\simeq 1 - \frac{K}{MT} + \frac{1}{2}\left(\frac{K}{MT}\right)^2 + O\left[\left(\frac{K}{MT}\right)^3\right], \end{aligned} \tag{2.5}$$

where in the second line we expanded the expression up to second order in the ratio $\frac{K}{MT} \ll 1$. We here use the big-O notation in the sense of infinitesimal asymptotics, which means that $O\left[\left(\frac{K}{MT}\right)^3\right]$ collects all terms of the form $f\left(\frac{K}{MT}\right)$ such that for any small $\frac{K}{MT}$ there exists a constant C fulfilling the relation $|f\left(\frac{K}{MT}\right)| < C\left(\frac{K}{MT}\right)^3$ as $\frac{K}{MT} \to 0$. In complexity theory big-O

often implicitly denotes the infinite asymptotics when it refers to an integer variable n. Both use cases of the notation are intended and comply with its definition (Knuth, 1997, section 1.2.11.1).

The expected number of target lists with at least one synapse is

$$\begin{aligned} N_c^{>0} &= (1 - p_\emptyset)\, N \\ &= \left(1 - \left(1 - \frac{1}{N}\right)^{\frac{KN}{MT}}\right) N. \end{aligned} \tag{2.6}$$

For the weak scaling shown in Fig. 2.1 we express $N = N_{\mathrm{VP}} MT$ in terms of the number of local neurons per virtual process N_{VP}. Using the definition of K_{VP}, Eq. 2.6 becomes

$$\tilde{N}_c^{>0} = \left(1 - e^{-\frac{K_{\mathrm{VP}}}{N}}\right) N. \tag{2.7}$$

In weak scaling the total number of local synapses K_{VP} remains constant and we find the limit of $N_c^{>0}$ by approximating the exponential to linear order

$$\lim_{N\to\infty} N_c^{>0} = \left(1 - \left(1 - \frac{K_{\mathrm{VP}}}{N}\right)\right) N = K_{\mathrm{VP}}.$$

Using Eq. 2.7 the number of neurons N_ζ at which a fraction ζ of the maximal number of target lists K_{VP} contains at least one synapse is given by the relation

$$\zeta K_{\mathrm{VP}} = \left(1 - e^{-\frac{K_{\mathrm{VP}}}{N_\zeta}}\right) N_\zeta. \tag{2.8}$$

With the substitution $s = -K_{\mathrm{VP}}/N_\zeta$ the relation is of the form $e^s = 1 + \zeta s$ and can be inverted using the Lambert-W function (Corless et al., 1996) yielding

$$N_\alpha = K_{\mathrm{VP}} \zeta \left[1 + \zeta W\left(-\frac{e^{-\frac{1}{\zeta}}}{\zeta}\right)\right]^{-1}. \tag{2.9}$$

Starting again from Eq. 2.8 with a second order approximation for the exponential

$$\zeta K_{\mathrm{VP}} \simeq \left(1 - \left(1 - \frac{K_{\mathrm{VP}}}{N_\zeta} + \frac{K_{\mathrm{VP}}^2}{2N_\zeta^2}\right)\right) N_\zeta$$

the relation depends linearly on N_ζ, so

$$N_\zeta \simeq \frac{K_{\mathrm{VP}}}{2\,(1-\zeta)}. \tag{2.10}$$

Following Eq. 2.4 the probability of a particular neuron to establish exactly one synapse with a

local neuron is

$$\begin{aligned} p_1 &= \left(1-\frac{1}{N}\right)^{KN/MT-1}\left(\frac{1}{N}\right)\frac{KN}{MT} \\ &\simeq e^{-\frac{K}{MT}}\frac{K}{MT} \\ &\simeq \left(1-\frac{K}{MT}\right)\frac{K}{MT}+O\left[\left(\frac{K}{MT}\right)^3\right]. \end{aligned} \tag{2.11}$$

Therefore the expected number of target lists with exactly one synapse is

$$\begin{aligned} N_{\mathrm{c}}^1 &= p_1 N = \left(1-\frac{1}{N}\right)^{K_{\mathrm{VP}}-1} K_{\mathrm{VP}} \\ &= \left(1-\frac{1}{MT\,N_{\mathrm{VP}}}\right)^{\frac{K}{MT}-1}\frac{K}{MT}. \end{aligned} \tag{2.12}$$

Naturally N_{c}^1 has the same limit K_{VP} as $N_{\mathrm{c}}^{>0}$. The probability to establish more than one synapse with a local neuron is the remainder

$$\begin{aligned} p_{>1} &= 1-p_\emptyset-p_1 \\ &= \left(1-\left(1-\frac{1}{N}\right)^{K_{\mathrm{VP}}}-\left(1-\frac{1}{N}\right)^{K_{\mathrm{VP}}-1}\left(\frac{1}{N}\right)K_{\mathrm{VP}}\right) \\ &\simeq 1-e^{-\frac{K}{MT}}-e^{-\frac{K}{MT}}\frac{K}{MT} \\ &\simeq \frac{K}{MT}-\frac{1}{2}\left(\frac{K}{MT}\right)^2-\left(1-\frac{K}{MT}\right)\frac{K}{MT}+O\left[\left(\frac{K}{MT}\right)^3\right] \\ &= \frac{1}{2}\left(\frac{K}{MT}\right)^2+O\left(\left(\frac{K}{MT}\right)^3\right), \end{aligned} \tag{2.13}$$

where from the second to the third line we ignored the -1 in the exponent, identified the exponential function in the limit, and from the third to the fourth line approximated the expression consistently up to second order in $\frac{K}{MT}$. The expected number of such target lists is

$$\begin{aligned} N_{\mathrm{c}}^{>1} &= \left(1-\left(1-\frac{1}{N}\right)^{K_{\mathrm{VP}}}-\left(1-\frac{1}{N}\right)^{K_{\mathrm{VP}}-1}\left(\frac{1}{N}\right)K_{\mathrm{VP}}\right)N \\ &\simeq \frac{1}{2}K_{\mathrm{VP}}^2\frac{1}{N}, \end{aligned}$$

which can be expressed in terms of MT and N_{VP} by noting that $N = N_{\mathrm{VP}}MT$ and $K_{\mathrm{VP}} = \frac{KN}{MT} = N_{\mathrm{VP}}K$. The limit exposes that the number of target lists with more than one synapse declines hyperbolically with N.

2.2.4 Supercomputers

The compute nodes in contemporary supercomputers contain multi-core processors; the trend towards ever greater numbers of cores is further manifested in the BlueGene/Q architecture with 16 cores per node, each capable of running 4 hardware threads. These architectures feature a multi-level parallel programming model, each level potentially operating at different granularity. The coarsest level is provided by the process based distribution, using MPI for inter-process communication (message passing interface, Message Passing Interface Forum, 1994). Within each process, the next finer level is covered by threads, which can be forked and joined in a flexible manner with OpenMP enabled compilers (Board, 2008). The finest level is provided by streaming instructions that make use of concurrently operating floating point units within each core.

To evaluate the scalability of NEST in terms of run time and memory usage we performed benchmarks on two different distributed-memory supercomputing systems: the JUQUEEN BlueGene/Q at the Jülich Research Centre in Germany and the K computer at the Advanced Institute for Computational Science in Kobe, Japan. The K computer consists of 88,128 compute nodes, each with an 8-core SPARC64 VIIIfx processor, which operates at a clock frequency of 2 GHz (Yonezawa et al., 2011), whereas the JUQUEEN supercomputer comprises 28,672 nodes, each with a 16-core IBM PowerPC A2 processor, which runs at 1.6 GHz. Both systems support a hybrid simulation scheme: distributed-memory parallel computing with MPI and multithreading on the processor level. In addition, the individual cores of a JUQUEEN processor support simultaneous multithreading with up to 4 threads. Both supercomputers have 16 GB of random access memory (RAM) available per compute node such that in terms of total memory resources the K computer is more than three times larger than JUQUEEN. The compute nodes of the K computer are connected with the "Tofu" (*to*rus connected *fu*ll connection) interconnect network, which is a six-dimensional mesh/torus network (Ajima et al., 2009). The bandwidth per link is 5 GB/s. JUQUEEN uses a five-dimensional torus interconnect network with a bandwidth of 2 GB/s per link.

In this study all benchmarks were run with $T = 8$ OpenMP threads per compute node, which on both systems results in 2 GB of memory per thread, and hence facilitates the direct comparison of benchmarking results between the two systems. With this setup we exploited all cores per node on the K computer but only half of the cores per node on JUQUEEN. In particular we did not make use of the hardware support for multithreading the individual processor cores of JUQUEEN already provide. In total on JUQUEEN only 8 of the 64 hardware supported threads were used.

2.2.5 Maximum-filling scaling

To obtain the maximum-filling scalings shown in Fig. 2.7–Fig. 2.11 we followed a two step procedure. First, based on the memory-usage model, we obtain a prediction of the maximum number of neurons fitting on a given portion of the machine. We then run a series of "dry runs", where the number of neurons is varied around the predicted value. The dry run is a feature of NEST that we originally developed to validate our model of the simulator's memory usage (Kunkel et al., 2012b). A dry run executes the same script as the actual simulation, but only uses one compute node. This feature can be enabled in NEST at run time by the simulation script. Due to the absence of the $M - 1$ other instances, the script can only be executed up to the point where the first communication takes place, namely until after the connectivity has been set up. At this point, however, the bulk of the memory has been allocated so that a good estimate of the resources can be obtained and the majority of the simulation script has been executed. In order to establish the same data structures as in the full run, the kernel needs to be given the information about the total number of processes in the actual simulation. This procedure also takes into account that of the nominal amount of working memory (e.g. 16 GB per processor on K) typically only a fraction (13.81 GB per processor on K) is actually available for the user program.

2.3 Results

2.3.1 Memory usage in the petascale regime

The kernel of NEST 2.2 (3g kernel) is discussed in detail in Kunkel et al. (2012b) and Helias et al. (2012). In Fig. 2.2 we compare the memory consumptions of the 3g kernel and the 4g kernel depending on the number of employed cores MT. We choose the number of neurons such that at each machine size the 4g kernel consumes the entire available memory. For the same network size, we estimate the memory consumption that the 3g kernel would require. The upper panel of Fig. 2.2 shows the different contributions to memory consumption for this earlier kernel. In the following we identify the dominant contributions in the limit of large machines used to guide the development of the 4g kernel. The resulting implementation of the 4g kernel is described in Sec. 2.3.2 to Sec. 2.3.5.

In simulations running $MT \sim 100$ (we use $\sim$ to read "on the order of") virtual processes, synapse objects take up most of the available memory. Hence, on small clusters a good approximation of the maximum possible network size is given by $N_{\max} \approx \mathcal{M}_{\max}/Km_{\mathrm{c}}$ where $\mathcal{M}_{\max}$ denotes the amount of memory available per MPI process. In the range of $MT \sim 1000$ virtual processes we observe a transition where due to an insufficient parallelization of data structures

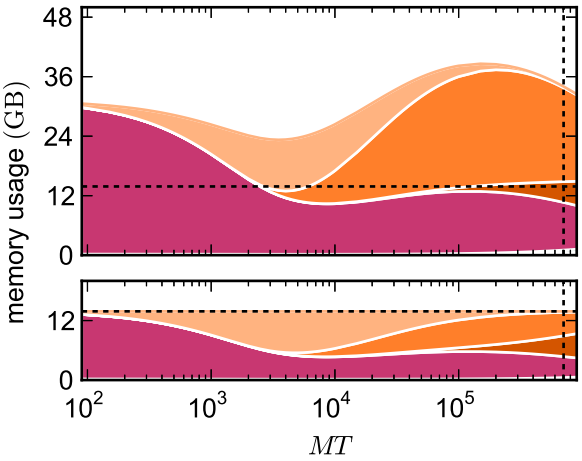

Figure 2.2: Predicted cumulative memory usage as a function of number of virtual processes for a maximum-filling scaling. Contributions of different data structure components to total memory usage $\mathcal{M}(M, T, N, K)$ of NEST for a network that just fits on MT cores of the K computer when using the 4g kernel with $T = 8$. Contributions of synapse objects and relevant components of connection infrastructure are shown in pink and shades of orange, respectively. Contributions of base memory usage, neuron objects, and neuron infrastructure are significantly smaller and hence not visible at this scale. $K = 11,250$ synapses per neuron are assumed. Dark orange: sparse table, orange: intermediate infrastructure containing exactly 1 synapse, light orange: intermediate infrastructure containing more than 1 synapse. Predicted memory usage is shown for 3g (upper panel) and 4g technology (lower panel) with identical scaling of the vertical axes. Vertical dashed black lines indicate full size of the K computer; horizontal dashed black lines indicate maximum memory usage measured on K.

the connection infrastructure (shades of orange) starts to dominate the total memory consumption. For sufficiently small machine sizes $MT <\sim 1000$ the intermediate synaptic infrastructure (shown in orange in Fig. 2.3A) typically stores on each virtual process more than one outgoing synapse for each source neuron. The entailed memory overhead is therefore negligible compared with the memory consumed by the actual synapse objects. As MT increases, the target lists become progressively shorter; the proportion of source neurons for which the target lists only store a single connection increases. We obtain a quantitative measure by help of the memory model presented in Sec. 2.3.1, considering the limit of very large machines where $K/(MT) \ll 1$. In this limit we can consistently expand all quantities up to second order in the ratio $\frac{K}{MT} \ll 1$. The probability Eq. 2.5 for a source neuron on a given machine to have an empty target list approaches unity $p_0 \to 1$. Correspondingly, the probability for a target list with exactly one entry Eq. 2.11 approaches $p_1 \to \frac{K}{MT}$. Target lists with more than one entry become quadratically unlikely in the small parameter $\frac{K}{MT}$ Eq. 2.13,

$$p_{>1} \simeq \frac{1}{2}\left(\frac{K}{MT}\right)^2 .$$

Two observations can be made from these expressions. In the sparse limit, the probabilities

become independent of the number of neurons. For small $\frac{K}{MT}$, target lists are short and, if not empty, typically contain only a single connection. To illustrate these estimates on a concrete example we assume the simulation of a network with $K = 10^4$ synapses per neuron distributed across the processors of a supercomputer, such as the K computer, with $M \simeq 80,000$ CPUs and $T = 8$ threads each. The above estimates then yield $p_\emptyset \simeq 0.984$, $p_1 \simeq 0.015$, and $p_{>1} \simeq 0.00012$. Hence, given there is at least one connection, the conditional probability to have one synapse is $\frac{p_1}{p_1+p_{>1}} \simeq 1 - \frac{1}{2}\frac{K}{MT} \simeq 0.992$ and the conditional probability to have more than one synapse is only $\frac{p_{>1}}{p_1+p_{>1}} \simeq \frac{1}{2}\frac{K}{MT} \simeq 0.008$.

Fig. 2.1 shows the number of non-empty target lists under weak scaling as a function of the network size N. In the limit of large networks this number approaches $\frac{NK}{MT}$, and is thus equal to the number of local synapses terminating on the respective machine. The size of the network N_ζ at which the number of target lists has reached a fraction $\zeta \simeq 1$ of the maximum number of lists is given by N_ζ Eq. 2.10 with $N_\zeta \simeq \frac{K_{\mathrm{VP}}}{2(1-\zeta)}$. The term depends linearly on the number of synapses per virtual process. A fraction of $\zeta = 0.95$ is thus already reached when the network size $N_\zeta \simeq K_{\mathrm{VP}}/(2 \cdot 0.05) = 10\,K_{\mathrm{VP}}$ exceeds the number of local synapses by one order of magnitude independent of the other parameters. This result is independent of the detailed parameters of the memory model, as it results from the generic combinatorics of the problem.

The effects on the memory requirements can be observed in Fig. 2.2 (top), where the amount of memory consumed by lists of length one and larger one are shown separately: at $MT \sim 10^4$ the memory consumed by the former starts exceeding the consumption of the latter. At $MT \sim 10^5$ the memory consumed by lists with more than one synapse is negligible. As we have seen in Sec. 2.2.4, the scenario at $MT \sim 10^5$ is the relevant one for currently available supercomputers. In the following we use the analysis above to guide our development of memory-efficient data structures on such machines. Fig. 2.2 (top) highlights that the intermediate synaptic infrastructure when storing only a single synapse must be lean. In the 3g kernel the memory consumed by a single synapse object is 48 B, while the overhead of the intermediate infrastructure is 136 B. Hence in the limit of sparseness, a synapse effectively costs 48 B + 136 B. Reducing the contribution of the intermediate infrastructure is therefore the first target of our optimizations described in Sec. 2.3.2. We identify the size of the synapse objects as the contribution of secondary importance and describe in Sec. 2.3.3 the corresponding optimization. The resulting small object sizes can only be exploited with a dedicated pool allocator (see Sec. 2.3.4). The least contribution to the memory footprint stems from the neuronal infrastructure, the improved design of which is documented in Sec. 2.3.5. The sparse table has an even larger contribution than the neuronal infrastructure. However, the employed collective communication scheme that transmits the occurrence of an action potential to all other machines requires the information whether or not the sending neuron has a target on a

particular machine. This information is represented close to optimal by the sparse table. In the current work we therefore refrained from changing this fundamental design decision.

2.3.2 Auto-adjusting connection infrastructure

Fig. 2.3A illustrates the connection infrastructure of NEST in the 3rd generation kernel (3g). As shown in Sec. 2.3.1 this data structure produces an overhead in the limit of virtual processes MT exceeding the number of outgoing synapses K per neuron; a presynaptic neuron then in most cases establishes zero or one synapse on a given core. The overhead can be avoided, because the intermediate data structure is merely required to distinguish different types of synapses originating from the same source neuron. For only a single outgoing synapse per source neuron it is not required to provide room to simultaneously store different types.

The main idea is hence to use data structures that automatically adapt to the stored information, as illustrated in Fig. 2.3B. The algorithm wiring the network then chooses from a set of pre-defined containers depending on the actual need. The corresponding data types are arranged in a class hierarchy shown in Fig. 2.4, with the abstract base class `ConnectorBase` defining a common interface. The wiring algorithm distinguishes four cases, depending on the number and types of the outgoing synapses of the given source neuron:

Case 0 The source neuron has no target on this machine. In this case, the unset bit in the sparse table (see Fig. 2.3) indicates the absence of synapses and no further data structures are created.

Case 1 The source neuron has outgoing synapses that are of the same type. In this case we use a type-homogeneous container `HomConnector`. Depending on the number of synapses, we use two different strategies to implement the homogeneous container. If less than K_{cutoff} synapses are stored, we employ a recursive C++ template definition of a structure that holds exactly $1, 2, \ldots, K_{\text{cutoff}}$ synapses. Here K_{cutoff} is a compile-time constant that throughout this work was chosen to be $K_{\text{cutoff}} = 3$. The recursive template definition is shown in Alg. 1 and follows the known pattern defining the recursion step with an integer-dependent template and the recursion termination by a specialization for one specific integer value (Vandervoorde & Josuttis, 2003, chapter 17). The classes are instantiated at compile time due to the recursive definition of the method `push_back`. The set of containers implements the functionality of a vector, requiring just the memory for the actual payload plus an overhead of 8 B for the virtual function table pointer due to the use of an abstract base class providing the interface of virtual functions. Our implementation uses a custom-made pool allocator ensuring that each thread allocates from

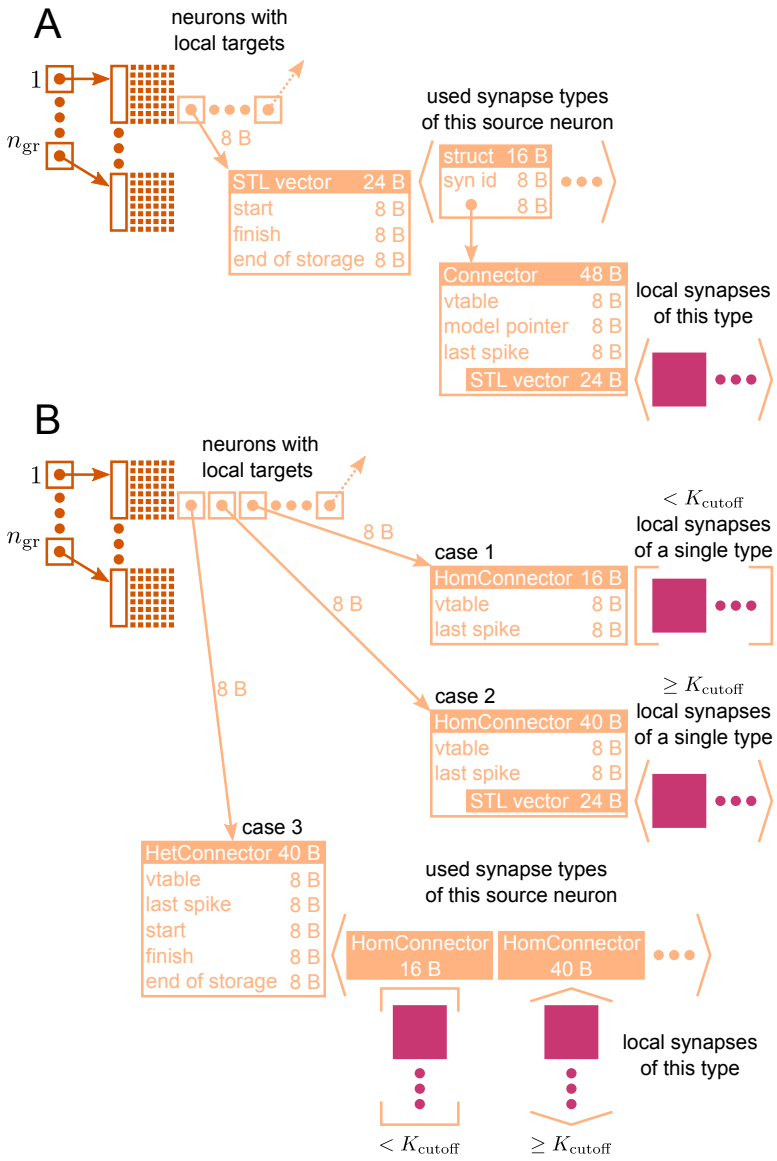

Figure 2.3: VP-local connection infrastructure of NEST. A sparse table (dark orange structure and attached light orange boxes with arrows) holds the thread-local connection objects (pink squares) sorted according to the global index of the associated presynaptic node. The sparse table consists of n_{gr} equally-sized groups, where each group maintains a bit field (tiny squares) with one bit for each global node index in the group indicating the presence or absence of local targets. If a particular node has local targets, the sparse table stores a pointer to an additional inner data structure (light orange), which has undergone a major redesign during the software development process that led from the 3g to the 4g kernel. **(A)** Connection infrastructure of the 3g kernel; listed Byte counts contribute to m_c^+ (see Eq. 2.3). The inner data structure consists of a `vector`, which holds a `struct` for each connection type that the node has locally in use. Each `struct` links the id of a particular connection type with a pointer to the `Connector` that stores the connection objects of this type in a `vector`. **(B)** Auto-adjusting connection infrastructure of the 4g kernel. Case 1: A particular node has less than K_{cutoff} local connections and all are of the same type. A lightweight inner structure (`HomConnector`) stores the connection objects in a fixed-size array. Listed Byte counts contribute to m_c^1. Case 2: A particular node has at least K_{cutoff} local connections and all are of the same type. A `HomConnector` stores the connection objects in a dynamically-sized `vector`. Case 3: The local connections of a particular node are of different types. A `HetConnector`, which is derived from C++ `vector`, holds a `HomConnector` (either case 1 or 2) for each connection type that the node has locally in use.

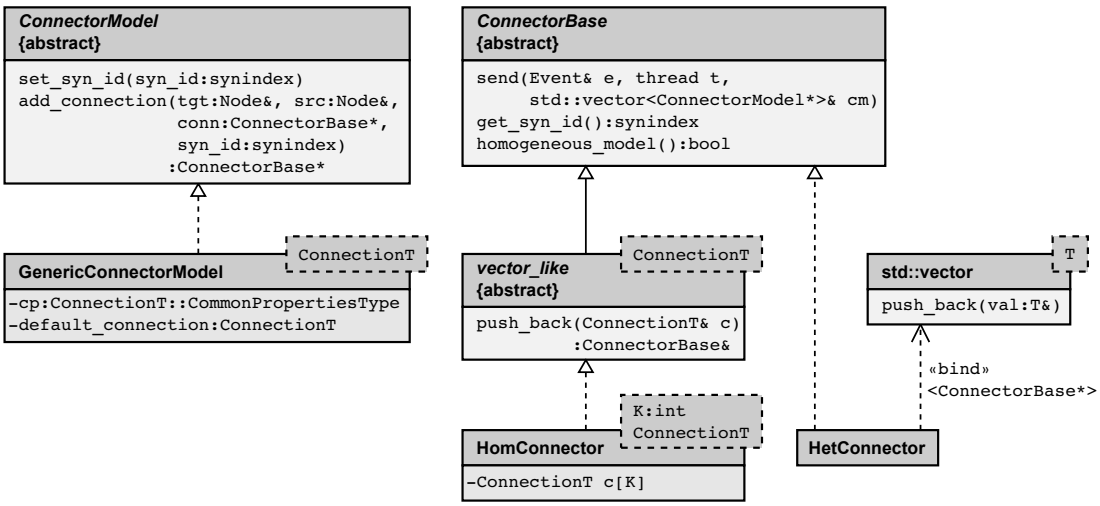

Figure 2.4: Simplified class diagram of the connection infrastructure. The base class `ConnectorModel` serves as a factory for connections. The member function `add_connection` is called when a new synapse is created and implements the algorithm to select the appropriate storage container (see also Alg. 2). The derived template `GenericConnectorModel` contains all code independent of synapse type and is instantiated once for each synapse type. It also holds the set of default values for synapse parameters (data member `default_connection`) and those parameters that are the same for all synapses of the respective type (data member `cp`). The class `ConnectorBase` defines the interface for all synapse containers, providing abstract members to deliver spike events (`send`), to determine the synapse type of type-homogeneous containers (`get_syn_id`), and to test if a container is type-homogeneous, i.e. whether it stores a single or several different synapse types (`homogeneous_model`). Two derived classes implement type-homogeneous containers (`HomConnector`) and type-heterogeneous containers (`HetConnector`). The latter, in turn, may contain several containers of class `HomConnector`, stored in a C++ standard library vector (by double inheritance from `std::vector`). `HomConnector` inherits the function `push_back` from the interface `vector_like`, required to implement the recursive sequence of container types, described in Alg. 1.

its own contiguous block of memory to improve the cache performance and to reduce the overhead of allocating many small objects (see Sec. 2.3.4).

Case 2 If more than K_{cutoff} synapses of the same type are stored we resort to the conventional implementation employing a `std::vector` from the C++ standard template library. The implementation of a vector entails an additional overhead of 3 times 8 B. This case provides the recursion termination for the set of homogeneous containers, as shown in Alg. 1.

Case 3 If a source neuron has synapses of different types targeting neurons on the same machine, we employ the container `HetConnector`. This intermediate container stores several homogeneous connectors (of either case 1 or 2 above) and is inherited from a `std::vector<ConnectorBase*>`.

The algorithm for creating new connections employing these adaptive data containers is documented as pseudo-code in Alg. 2.

2.3.3 Condensed synapse objects

In a typical cortical neuronal network model, the number of synapses exceeds the number of neurons by a factor $\sim 10^4$. To reduce the memory consumption of a simulation, it is thus most efficient to optimize the size of synapse objects, outlined in our agenda at the end of Sec. 2.2.2 as step two of our optimizations. We therefore reviewed the `Connection` class (representing the synapses) and identified data members that could be represented more compactly or which could even be removed. In refactoring the synapse objects our objective is to neither compromise on functionality nor on the precision of synaptic state variables, such as the synaptic weight or the spike trace variables needed for spike-timing dependent plasticity (Morrison et al., 2007); these state variables are still of type `double` in the new simulation kernel (4g). Our analysis of the `Connection` data structure identified three steps to further reduce the memory consumption of single synapse objects. The steps are explained in the following three subsections and concluded by a subsection summarizing the resulting reduced memory footprint.

Avoidance of polymorphic synapse objects

As shown in Fig. 2.4 and in Alg. 1, the container-classes are templates with the synapse type `connectionT` as a template parameter. Consequently, the container itself does not need a polymorphic interface, because by specialization for a particular synapse type this type is known and fixed at compile time. The only exception to this rule is synapse creation: we here need to check that the synapse model and the involved neuron models are compatible. More precisely,

we need to ensure that (i) the new connection and (ii) the target node are able to handle the type of events sent by the source node. In NEST 2.2 (3g) the first of the two checks (i) requires a common interface to all synapse objects (i.e. an abstract base class `Connection`) that provides a set of virtual functions, one for each possible event type (spike events, voltage measurement requests, etc.). The synapse object then implements only those virtual member functions for event types it can handle. A similar pattern is used for the check (ii), determining whether the target node is able to handle the incoming event type. On a 64 bit architecture the virtual base class causes a per-object overhead of 8 B for the virtual function table pointer. Having such a pointer in each node hardly affects the memory consumption, but spending these additional 8 B per synapse object seems rather costly given the fact that the connection handshake is the only instance when NEST exploits the polymorphism of `Connection`. Therefore, in the 4g kernel we redesigned the handshaking algorithm such that it still makes use of the polymorphism of `Node` but no longer requires virtual functions in `Connection`. This reduces the per-synapse memory usage m_c by 8 B.

The design pattern that circumvents polymorphic synapse objects is derived from the visitor pattern (Alexandrescu, 2001; Gamma et al., 1994). A sequence diagram of the connection setup is shown in Fig. 2.5. The crucial step is to shift the set of virtual functions that check the validity of received events from the synapse objects to a nested class, called `check_helper`. Each connection class owns its specific version of `check_helper`, which is derived from the `Node` base class. This inner class redefines the virtual function `handles_test_event` for those event types the connection model can handle. The default implementations inherited from the base class throw an exception and thus by default signal the inability of the synapse to handle the particular event. Since the connection class only contains the nested class definition, rather than a member of type `check_helper`, the nested class does not contribute to the memory footprint of the connection.

This new design has the additional advantage that checks (i) and (ii) have the same structure following the visitor pattern, shown in Fig. 2.5: For check (i), the synapse creates an object of its corresponding type `check_helper`, passes it as a visitor to the source neuron's member function `send_test_event`, which in turn calls the overloaded version of the virtual function `handles_test_event` that has the matching event type. The test passes if `handles_test_event` is implemented for the type of event `e`, shown for the example of a `SpikeEvent` in Fig. 2.5. The second check (ii) proceeds along analogous lines. Here the target neuron is passed as a visitor to the source neuron's member function `send_test_event`, which in turn sends the event via a call to the target neuron's `handles_test_event` method.

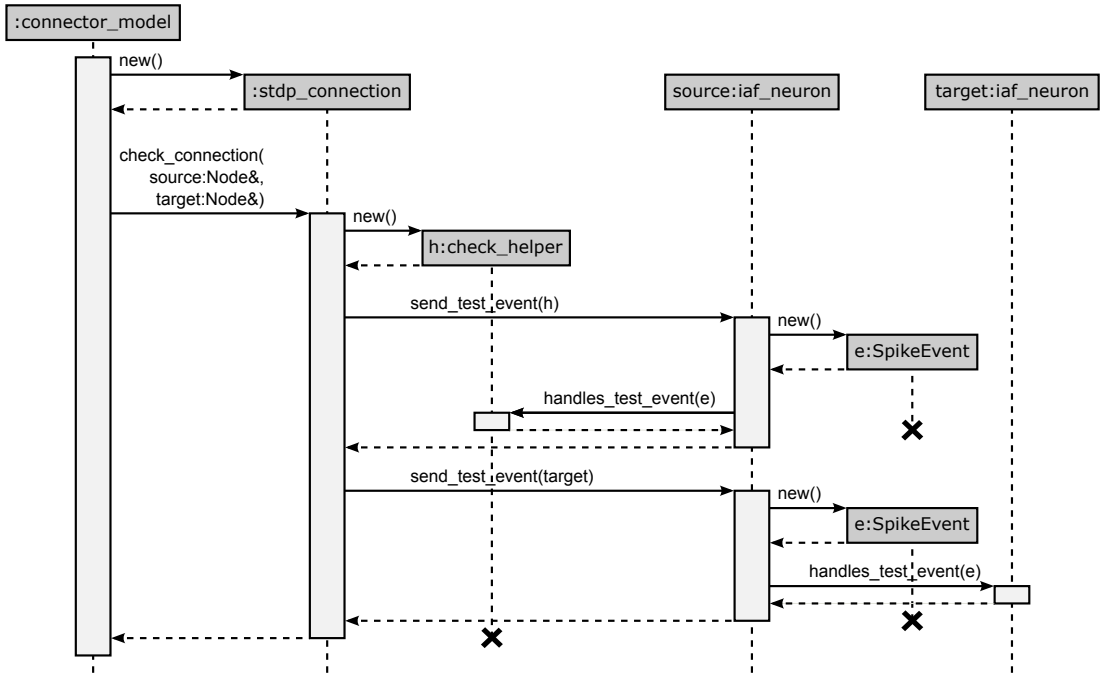

Figure 2.5: Connection handshaking mechanism of the 4g kernel. By executing connect, NEST calls `add_connection` of the corresponding connector in the connection infrastructure. This function creates a new instance for the connection, sets its parameters and starts the connection handshaking by calling `check_connection`. This function creates an instance of the `check_helper` class and calls `send_test_event` of the source node twice to send test events to the synapse represented by `check_helper` and to the target node, respectively. Both instances execute `handles_test_event` which, if the event cannot be handled, ends in the base-class implementation throwing an exception.

Indexed target addressing

A connection has to store the information about its target node. In NEST 2.2 (3g) this was solved by storing a pointer to the target node, consuming 8 B on a 64 bit architecture. The new simulation kernel (4g) implements the target addressing flexibly via a template argument `targetidentifierT` to the connection base class. This template parameter enables the formulation of synapse objects independent of how the target is stored, as long as the `targetidentifierT` provides methods to obtain the actual target neuron's physical address via a member `get_target_ptr()`. Besides the original implementation that stores the target as a full pointer, the 4g kernel supports indexed addressing of the target neuron. The two addressing schemes correspond to different implementations of the target identifier. Each synapse type may be instantiated with either of them.

Indexed addressing makes use of the limited number of neurons that are local to a given virtual process. These nodes are stored in a vector of pointers required to update the neuronal dynamics. In a parallel simulation, the number of local neurons rarely exceeds $\sim 10^4$ nodes. The index space of local nodes is thus sufficiently covered by indices of 2 B length corresponding

to a maximal index of $2^{16} - 1 = 65,535$. The target identifier then stores the index of the target neuron corresponding to its position in the vector of VP-local nodes. We call this index "thread-local id" in the following. Determining the target requires one more indirection to compute the actual address from the stored index, but saves 6 B of memory per target pointer. The implementation requires an extension of the user interface, because in the 3g kernel the thread-local vectors of nodes are only generated dynamically at the start of the simulation. The translation from the global id (GID) of a target neuron to the thread-local id is hence unavailable during wiring of the network . To employ the indexed addressing scheme the user therefore needs to execute the function `CreateThreadLocalIds` after all nodes have been created and prior to the creation of any synapse using the indexed addressing scheme. This function fills the vectors holding the thread-local neurons and moreover stores the thread-local id for each neuron in its base class. The latter information is needed during connection setup to obtain the thread-local id from a given global neuron id.

Combined storage of delay and synapse type

In a third step, the storage of the synapse type (`syn_id`) and the delay (d) of a connection are optimized. The 3rd generation connections stored these properties separately, the delay as a long integer variable with 8 B and the synapse type as an unsigned integer with 4 B length. The delay is represented as an integer multiple of the chosen simulation resolution h, typically $h = 0.1\,\text{ms}$ (Morrison et al., 2005). A reduction of the storage size from 8 B to 3 B hence corresponds to a restriction of the maximum delay (assuming $h = 0.1\,\text{ms}$) from $2^{64}\,h \simeq 1.4\,10^9\,\text{years}$ to $2^{24}\,h \simeq 1670\,\text{s}$, which is more than sufficient for all practical demands. The limitation of the synapse type identifier to an unsigned int of 1 B limits the number of different synapse types to $2^8 = 256$, a reasonable upper limit for different synapse models. To ensure that delay and synapse identifier are stored as compactly as possible independent of memory alignment choices of the compiler, both variables are stored as 8 bit and 24 bit fields (Stroustrup, 1997, chapter C.8.1) in a common structure requiring 4 B; the structure provides methods for convenient access to synapse id and delay. Overall, the new storage scheme thus requires just 4 B per synapse instead of 8 B + 4 B required by the 3rd generation kernel for the same information, saving 8 B of memory.

Memory footprint of synapse objects

The simplest type of synapse, the static synapse, is described completely by information identifying its target, type, delay and weight. In the 3g simulation kernel, the type required 4 B, while the three other items each required 8 B as described above, for a total of 28 B. On computers with 64 bit architecture, data types requiring 8 B, such as pointers, doubles and long integers,

are by default aligned with 64 bit-word boundaries, so that a static synapse object in effect required 32 B: the 4 B unsigned integer representing the synapse type was padded to 8 B to ensure proper alignment of the following 8 B variable.

Combining all three 4g kernel optimizations described above, the size of a static synapse object shrinks to 2 B for the target index, 4 B for the combined synapse type id and delay, and 8 B for the synaptic weight (double precision). Provided that the data members in the synapse object are ordered as is {`target index, synapse type id + delay, synaptic weight`}, the object requires 16 B in total, including 2 B of padding inserted after the target index.

In addition to the member variables of a static synapse, an object representing an STDP synapse stores the synaptic trace variable (double) (Morrison et al., 2008), consuming another 8 B. The total memory footprint for STDP synapse objects is thus 24 B, of which 22 B or 92% hold data, while the remaining 8% are padding. On some systems, the 2 B of padding might in principle be avoided by forcing the compiler to generate code for unaligned memory access, e.g., using the `__attribute__(packed)` provided by the GNU C++ compiler (Manual & Stalhnan, 2013, chapter 6.36). Unfortunately, such packing is not defined in the C++-standard and thus compiler-dependent; on some architectures, unaligned memory access may also incur significant runtime overhead (Rentzsch, 2005). Given that synapse objects contribute less than 50% to the total memory requirement of the simulator for large networks, as shown in Fig. 2.2, the memory overhead incurred by padding is thus less than 4%. To ensure maximum portability, NEST does not use unaligned memory access.

2.3.4 Pool allocator

The standard allocator on most systems has two severe problems with regard to memory locality when using multiple threads and memory overhead when allocating many small objects. The first problem stems from the allocator not taking into account the actual physical layout of the memory architecture of the machine. In particular, the memory for objects of different threads is often not separated, but objects are rather allocated in the order in which they are created. This makes caching on multi-core machines with different caches for different cores or with different memory banks for different cores (cf. ccNUMA) inefficient and leads to frequent (and slow) cache reloads from the main memory, a problem generally referred to as cache thrashing. The second problem is caused by the fact that the allocator needs to keep administrative information about each single allocation for freeing the memory and returning it to the operation system after its use. Usually, this information consists at least of a pointer to the data (8 B) and the size of the allocated memory (8 B). If the size of administrative data is in the range of the size of the allocation, this means a significant memory overhead.

To ameliorate these problems, we implemented a stateless custom allocator, which does not keep any administrative information about allocations and provides thread-local memory pools for the storage of connection containers (see Stroustrup, 1997, chapter 19.4.2 for the basic concept). The absence of allocation information is not a problem, as the data structures for the storage of synapses only grow, and never are freed during the run of the program. Moreover, as shown in Sec. 2.3.1, in the sparse limit reallocation of target lists is rare, as most target lists contain only a single synapse. In this limit the simple pool allocator is hence close to optimal. The use of the pool allocator needs to be selected by the user with a compile-time switch. On small clusters or desktop machines, where frequent reallocation takes place, the standard allocator is recommended.

2.3.5 Sparse node array

Kunkel et al. (2012b) showed that the memory overhead required to represent neurons on each MPI process could be reduced considerably by using a sparse table. In such a table, each non-local neuron is represented by a single bit, while the overhead for local neurons is only a few Bytes. Kunkel et al. (2012b) give the following expression for the memory required to represent neurons:

$$\mathcal{M}_{\mathrm{n}}\left(M,N\right)=Nm_{\mathrm{n}}^{0}+\left(N-\frac{N}{M}\right)m_{\mathrm{n}}^{\emptyset}+\frac{N}{M}\left(m_{\mathrm{n}}^{+}+m_{\mathrm{n}}\right). \tag{2.14}$$

Using the sparse table, one has the following parameters: $m_{\mathrm{n}}^{0}=\frac{1}{3}\,\mathrm{B}$, $m_{\mathrm{n}}^{\emptyset}=0\,\mathrm{B}$, $m_{\mathrm{n}}^{+}=24\,\mathrm{B}$ and $m_{\mathrm{n}}\approx 1000\,\mathrm{B}$, where the first three parameters describe overhead, while the last parameter represents actual neuron objects. With $N/M\sim 10^{3}$ neurons per MPI process, neuron objects consume $\sim 1\,\mathrm{MB}$, while the overhead from Equation Eq. 2.14 is

$$\mathcal{M}_{\mathrm{n}}^{\mathrm{overhead}}\left(M,N\right)=N\times\frac{1}{3}\,\mathrm{B}+\frac{N}{M}\times 24\,\mathrm{B}. \tag{2.15}$$

For $N/M\sim 10^{3}$, the second term is about 24 kB and thus negligible, while the first term becomes appreciable for large networks. Indeed, for $N=10^{8}$ neurons, it amounts to approximately 32 MB, for $N=10^{9}$ to 318 MB, almost all of which are consumed for zero-bits in the bit array of the sparse table. This not only requires appreciable amounts of memory to represent vanishing amounts of information, it also means that sparse table lookups, which in most cases return negative results, become cache inefficient, as the size of the bit array by far exceeds the cache size.

In the following we show how this overhead can be eliminated entirely. The basic idea is to exploit the round-robin distribution of neurons to virtual processes and thus MPI processes

(Plesser et al., 2007; Eppler, 2006). As NEST builds a network, neurons are created with strictly increasing global neuron ids (GIDs) g, and each neuron is assigned to and stored on MPI rank

$$m_g = g \mod M \,. \tag{2.16}$$

Thus, if we place the neurons assigned to a single MPI process in an array in order of creation, GIDs of neighboring neurons will differ by M. For any given g, we thus can use linear interpolation between the GIDs of the first and last local neuron to determine the index pertaining to that GID in the array of local neurons. We then only need to check whether the neuron found at that index has the correct GID (i.e., is local to the MPI process) or not, in the latter case we conclude that the neuron is managed by a different MPI process.

Unfortunately, reality is slightly more complicated. Certain network elements are replicated on all virtual processes in NEST, either of logical necessity (subnet nodes used to structure large networks) or for performance reasons (stimulating and recording devices); we refer to such network elements as replicated nodes. This means that (i) any node g is guaranteed to be local to MPI rank m_g according to Eq. Eq. 2.16, (ii) if node g is a replicating node, it is also local to all other MPI ranks, and thus (iii) neighbors in the array of nodes are not necessarily spaced in intervals of M. This will skew the linear interpolation used to look up nodes by their GIDs.

Fortunately, the number of replicating nodes must be small in any practical simulation, because the memory requirement of the simulation would not scale otherwise. Thus, the skew will be small. This suggests the following approach: Let n_{loc} be the number of local nodes for a given MPI rank and $g_{\text{loc,min}}$ and $g_{\text{loc,max}}$ the smallest and largest local GID, respectively (we ignore that the root network with GID 0 is local to all ranks), and define the scaling factor

$$\alpha = \frac{n_{\text{loc}} - 2}{g_{\text{loc,max}} - g_{\text{loc,min}}} \,. \tag{2.17}$$

Then

$$l_g^* = \lfloor 1 + \alpha(g - g_{\text{loc,min}}) \rfloor \tag{2.18}$$

provides a linear estimate of the index of the node with GID g in the local array of nodes. This estimate is exact if we have a single MPI process or there are no replicating nodes except the root network. We can thus look up nodes by starting our search at index l_g^* in the array of local nodes and then proceed to smaller or larger nodes until we either have found the node with GID g or reached a node with GID smaller (larger) than g, in which case we conclude that node g is not local. The complete algorithm is given as Alg. 3.

The underlying data structure is the `SparseNodeArray`: Its main component is a C++

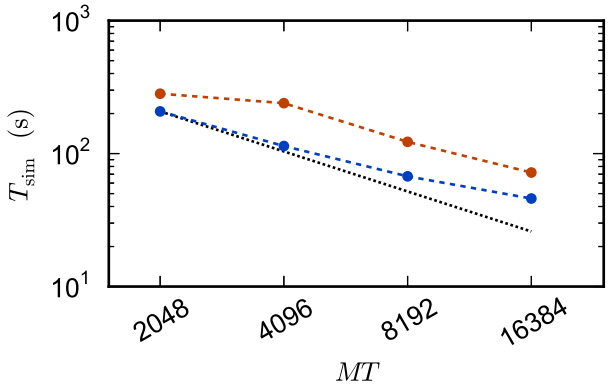

Figure 2.6: Strong scaling on K computer and JUQUEEN. Solid curves show the scaling of simulation time, dashed curves show setup time for networks of $N = 5,242,880$ neurons on JUQUEEN (blue) and $N = 5,210,112$ on the K computer (red). Black dotted lines are the linear expectations for simulation and setup. All simulations were carried out using the parameters of set 1 (cf. Sec. 2.2.1).

`vector` that stores in each element the pointer to a node and the GID of that node. This ensures that the linear search from l_g^* is cache efficient, compared to a `vector` keeping node pointers only, which would require looking up the GID via the node pointer in each search step. Additionally, the `SparseNodeArray` keeps track of the largest GID in the entire network (to detect invalid GIDs), the smallest and largest local GID and the scaling factor α. The memory overhead for this data structure is thus one pointer and one long per *local* node, i.e.,

$$\mathcal{M}_{\mathrm{n}}^{\mathrm{overhead}}(M, N) = \frac{N}{M} \times 16\,\mathrm{B}. \tag{2.19}$$

The overhead now only depends on the number of neurons per process, N/M, but no longer directly on the total number of neurons in the network, N.

To evaluate the quality of the linear estimate provided by Eq. 2.18, we counted the number of search steps required to find the correct node using a specially instrumented version of the NEST code. We collected data for $M = 2048$ to $M = 65,536$ MPI processes with four threads each and between 550 and 230 nodes per thread. The average number of search steps is 0.7 in this data and lookup never requires more than two steps for over 3 billion analyzed lookups. Therefore, we consider the linear estimate to be close to optimal. In addition to the lookup by GID, the `SparseNodeArray` also provides an interface for direct iteration solely over the local nodes to allow for, e.g., efficient initialization of all nodes.

2.3.6 Performance of 4g technology

Fig. 2.6 shows a strong scaling for the 4g kernel. At the left-most point the workload per core is maximal; the number of neurons is chosen such that the memory is completely filled. Near

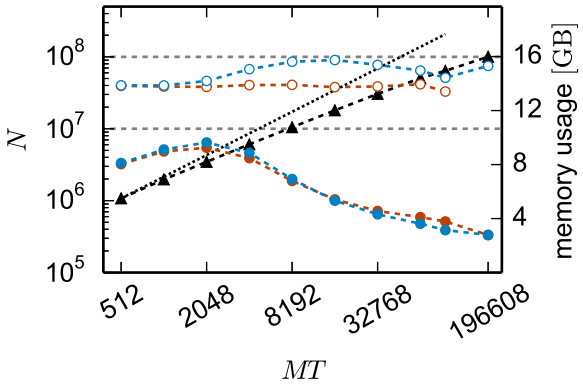

Figure 2.7: Comparison of memory usage of 3g and 4g technology. Black triangles show the maximum possible network size, using parameters of set 1 (cf. Sec. 2.2.1), that can be simulated on the K computer and on JUQUEEN when using the 3g technology with $T = 8$ threads per compute node (left vertical axis). The dotted black line indicates the ideal case where the network size increases with the same factor as the number of virtual processes. Circles show the memory consumption (right vertical axis) of simulations with the 3g (open) and 4g (filled) technology on the K computer (red) and JUQUEEN (blue), respectively.

the point of maximum workload the scaling on JUQUEEN is close to optimal, but degrades for lower workload. This is expected as ultimately the serial overhead dominates the simulation time. The relevance of a strong scaling graph is therefore naturally limited. To minimize the queuing time and the energy consumption it is desirable to choose the smallest possible machine size that enables the simulation of a given problem size. In the following we will hence study maximum-filling scalings, where for a given machine size MT we simulate the largest possible network that completely fills the memory of the machine. The procedure to arrive at the maximum possible network size is described in Sec. 2.2.5.

The reduction of the memory consumption of the 4g kernel compared to the 3g kernel is shown in Fig. 2.7. At a given machine size MT the same network is simulated with both kernels. For each MT the size of the network is chosen such that the 3g simulation consumes all available memory on the K computer (maximum-filling scaling). At high numbers of cores around $MT \sim 100,000$ the 4g kernel reduces the required memory by a factor of more than 3, for smaller machine sizes the reduction of memory consumption is less, but still substantial.

Fig. 2.8 shows the time to setup the network (top panel) and the simulation time (bottom panel) for the 3g and the 4g kernel in the same maximum-filling scaling as in Fig. 2.7. Although both kernels simulated the same network at a given MT and hence the same computation takes place in both simulations (same update steps of neurons and synapses, same activity, etc.), the run time of the new kernel is typically reduced especially at large machine sizes. On JUQUEEN this reduction monotonically increases with machine size, on the K computer the simulation time exhibits fluctuations that presumably originate from different load levels caused by other users of the machine, potentially occluding a clear monotonic dependence. The faster simulation

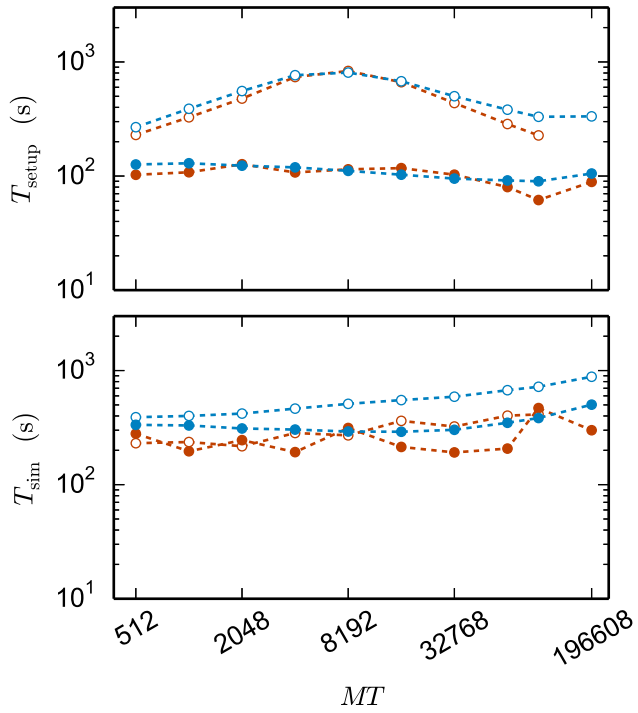

Figure 2.8: Comparison of performance of 3g and 4g technology. Setup time (upper panel) and simulation time (lower panel) as a function of number of virtual processes for the maximum network size that can be simulated when using the 3g technology (see Fig. 2.7) and parameter set 1 (cf. Sec. 2.2.1). Circles refer to simulations with 3g (open) and 4g (filled) technology, respectively. Results are shown for the K computer (red) and JUQUEEN (blue) using $T = 8$ threads per compute node.

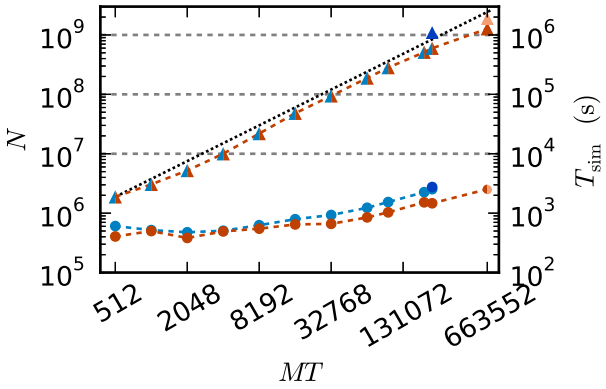

Figure 2.9: Maximum network size and corresponding run time as a function of number of virtual processes. Triangles show the maximum network size that can be simulated with parameter set 1 (cf. Sec. 2.2.1) on the K computer (red) and on JUQUEEN (blue) when using the 4g technology with $T = 8$ threads per compute node. The dotted black line indicates the ideal case where the network size increases with the same factor as the number of virtual processes. Dark blue (JUQUEEN) and orange (K) triangles represent the maximum network size using all compute nodes and parameter set 2 (JUQUEEN: $1.08 \cdot 10^9$ neurons, K: $1.86 \cdot 10^9$ neurons). Filled circles show the corresponding wall-clock time required to simulate the network for 1 s of biological time.

time of the new kernel points at the random memory access as an important contribution to the computation time. The smaller objects of the connection infrastructure enable more efficient use of the cache and reduce the overall required memory bandwidth.

The 4g implementation exhibits a reduction in setup time by a factor of 2–8 depending on network size (Fig. 2.8, top panel). In parts this higher performance is due to ongoing conventional optimization of the wiring routines; for example, by representing consecutive neuron ids within the wiring process by the beginning and end of the range (4g) instead of by explicitly naming all elements (3g). In parts the difference is due to faster memory allocation through the dedicated pool allocator and the smaller objects representing synapses and connection infrastructure.

The comparison of the 3g and 4g kernels in Fig. 2.7 and Fig. 2.8 is based on the maximum network size the 3g kernel can represent on a given number of cores. The reduced memory consumption of the 4g kernel allows us to simulate larger networks with the same computational resources. In Fig. 2.9 we therefore show a maximum-filling scaling determined for the 4g kernel, showing the maximum network size that can be simulated with a given number of cores MT and hence for a given amount of working memory. The growth of network size N with MT stays close to the ideal line.

Using parameter set 2 with $K = 6000$ synapses per neuron and employing all $82,944$ nodes of the K computer simultaneously in a single simulation with 8 cores each, we reached a maximum network size of $1.86 \cdot 10^9$ neurons and a total of $11.1 \cdot 10^{12}$ synapses. This is the largest simulation of a spiking neuronal network reported so far (RIKEN BSI, 2013). This world-record

simulation was performed on K, because only this machine provides the necessary memory to represent all synapses of the simulation with generic connectivity. Access to JUQUEEN and its predecessor JUGENE was, however, crucial for the design and implementation of the simulation kernel during the development phase of the K computer and for performing smaller simulations testing the implementation (Diesmann, 2012). In terms of memory, the K computer is at the time of writing the second largest computer (1.4 PB RAM, on the nodes available here 1.3 PB), exceeded only by the IBM sequoia computer (1.5 PB) at the Lawrence Livermore National Laboratory. JUQUEEN provides about one-third (0.46 PB) of the memory of the former two systems. Previous to the current report, the largest spiking network simulation comprised $1.62 \cdot 10^9$ neurons with on average about 5700 synapses per neuron (Ananthanarayanan et al., 2009). The simulation required less than 0.144 PB of memory by making use of a specific modular connectivity structure of the network. Thus, in contrast to the case of an arbitrary network discussed in the present study, the choice of a specific structure enabled the authors to condense the memory components of the connectivity infrastructure, corresponding in our implementation to the sparse table, the intermediate infrastructure, and the synapses (see Fig. 2.2), into effectively only 16 B per synapse.

Comparing the theoretical prediction of the memory model to the empirically found maximum network size reveals that the theory underestimates the actual memory consumption. As shown in our previous work (Kunkel et al., 2012b) these deviations are most likely due to underestimation of the object sizes. A direct comparison of the memory consumption of n objects of size m to the predicted value nm typically uncovers non-optimal object alignment. Obtaining instead the effective object sizes by linear regression, as in our earlier work (Kunkel et al., 2012b), would decrease the deviation of the model from the empirical result.

Comparing the memory resources required by the 3g and 4g kernels as a function of the just fitting network size (Fig. 2.10) shows the new kernel to be closer to the optimal linear scaling: doubling the machine size nearly doubles the network size that can be simulated. At $MT = 196,608$ virtual processes, the maximum network size possible with the 4g kernel is $N^{4g}_{max} = 5.1 \cdot 10^8$ compared to $N^{3g}_{max} = 1.0 \cdot 10^8$ for the 3g kernel. The absolute simulation time increases in the maximum-filling scheme for the 4g kernel compared to the 3g kernel, because the higher number of neurons per core constitutes a proportionally larger workload per core. For a fair comparison we therefore show in Fig. 2.11 the product of the runtime and the number of cores. The 4g technology shows a smaller slope of the dependence of the required resources on network size. This quantity is also of interest for the estimation of resources when planning research projects and applying for the required computation time as core hours are a commonly used unit in such documents. The duration of the simulation used for these benchmarks is 1 s of biological time. For longer times the core hours can be multiplied by the corresponding

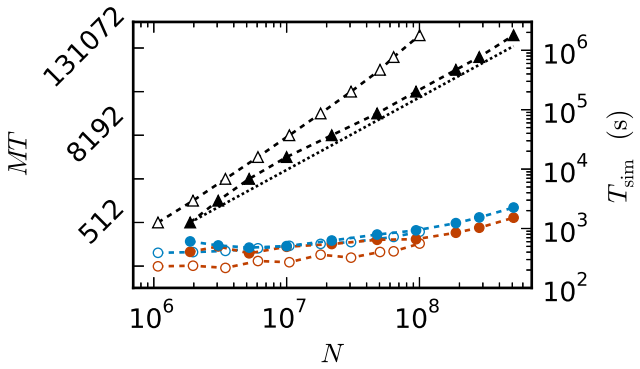

Figure 2.10: Comparison of maximum-filling scaling for 3g and 4g. Required number of cores MT (black, left vertical axis; numbers quadrupling between ticks) and time (colors, right vertical axis) as a function of network size for a maximum filling with parameter set 1 (cf. Sec. 2.2.1) for the 3g kernel (open symbols) and the 4g kernel (filled symbols). Benchmarks on the K computer shown in red, on JUQUEEN in blue. The dashed black line indicates the ideal case where the required number of cores increases only with the same factor as the network size.

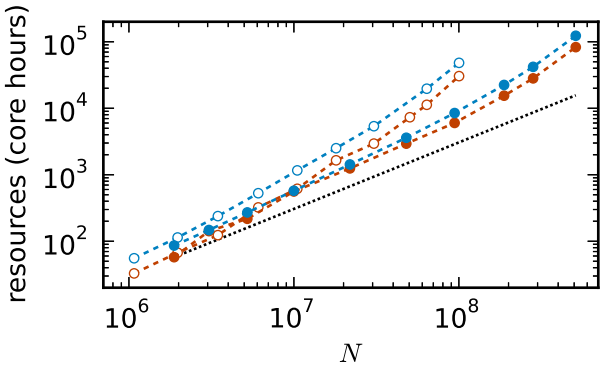

Figure 2.11: Required resources to simulate a network of size N. core hours are defined as $MT \times T_{\mathrm{sim}}$ and do not account for idle cores of a node. Filled circles correspond to the 4g technology (see Fig. 2.9) and open circles to the 3g technology (see Fig. 2.7 for maximum network size and lower panel of Fig. 2.8 for core hours). Results for K computer (red) and JUQUEEN (blue) using parameter set 1 (cf. Sec. 2.2.1). The dashed black line indicates the ideal case where the required resources increase with the same factor as the network size.

factor to obtain an estimate of the required resources. For example, given the resources of $10,000$ core hours on the K computer we can perform a simulation of one second of biological time with about $7 \cdot 10^7$ neurons using the 3g kernel but with more than twice as many neurons ($1.5 \cdot 10^8$) using the 4g kernel.

2.4 Conclusion

In this chapter, we have described the fourth generation kernel of the NEST simulation software. We started by explaining the necessity of this work by analyzing the limitations of the third generation kernel, which scales well up to $\sim 10^4$ compute cores. The reason for this limitation is that in a cortical network with a typical number of outgoing synapses of $\sim 10^4$ synapses, each neuron will probably have zero or one targets with a unique synapse type on each virtual process for machine sizes $\gg 10^4$. The NEST 3g connection infrastructure, designed for storing long target lists and heterogeneous synapse types, then produces unnecessary memory overhead which prevents a good weak scaling of the software. The new kernel, NEST 4g, implements a slim infrastructure for the described case of large networks on 10^5 and more virtual processes and effectively removes memory overhead. With its adaptive connection infrastructure, the kernel dynamically adjusts to the needs of the network to be simulated and thus maintains its generability on computer systems from laptops to supercomputers. In combination with three other optimizations, namely the condensed synapse objects, the pool allocator and the sparse node array, this leads to a drastic improvement in performance. We assessed the performance by performing strong scaling and maximum filling benchmark tests with a random E-I network, featuring random connectivity between neurons. The simulations demonstrated a reduction of the memory footprint on a supercomputer by a factor of four at large machine sizes and a significant decrease of the time needed to create the network. The benefit is twofold: On the one hand, this increases the size of the maximally representable network for a given machine size, leading to the world-record simulation with $1.86 \cdot 10^9$ neurons and $11.1 \cdot 10^{12}$ synapses on the K computer (RIKEN BSI, 2013). On the other hand, the required machine size for a given network is reduced, enabling the researcher to save resources.

CHAPTER 3

Construction of the spiking network model

The following chapter has been published as preprint in Schmidt M, Bakker R, Shen K, Bezgin G, Hilgetag C-C, Diesmann M, and van Albada SJ (2015). Full-density multi-scale account of structure and dynamics of macaque visual cortex. arXiv:1511.09364v2 (Schmidt et al., 2015a). Preliminary results have been presented in Schmidt et al. (2013) and Schmidt et al. (2014).

Author contributions: Under supervision of Sacha J. van Albada, Rembrandt Bakker and Markus Diesmann, the author performed the literature research, the conceptual work on all parts of the model definition and the implementation of the concepts.

3.1 Introduction

The model comprises 32 areas of the macaque cortex involved in visual processing in the parcellation of Felleman & Van Essen (1991), in the following referred to as FV91 (Table 6.1). Each area consists of 8 populations representing layers 2/3, 4, 5 and 6 each with an excitatory and an inhibitory population, with the exception of area TH which does not contain a granular layer. The model for each area is based on that of Potjans & Diesmann (2014), which we refer to as *the microcircuit model*. Table 3.1 summarizes the resulting multi-area model, in which each area is represented by a $1\,\mathrm{mm}^2$ patch of cortical surface.

The single-neuron dynamics and the distributions of intra-areal delays correspond to those of Potjans & Diesmann (2014). The neuron and synapse parameters are listed in Table 6.3. We assume a constant transmission speed of $v_\mathrm{t} = 3.5\,\mathrm{m/s}$ between areas as measured in monkey visual cortex (Girard et al., 2001), and draw the inter-areal delays from a Gaussian distribution with mean $d = s/v_\mathrm{t}$ where s is the distance, and standard deviation $\delta d = d/2$. The distance between each pair of areas is computed as the median of the distances between all vertex pairs of these two areas in their surface representation in F99 space, a standard macaque cortical surface included with Caret (Van Essen et al., 2001), where the vertex-to-vertex distance is the length of the shortest possible path without crossing the cortical surface (Bojak et al., 2011). Table 3.1 summarizes the components of the model.

In the following, we describe the data integration process leading to population sizes and connection probabilities from a combination of experimental data and statistical regularities.

3.2 Area-specific laminar compositions

First, we estimate the number of neurons $N(A, i)$ in population i of area A in three steps:

1. Neuronal volume densities provided in a different parcellation scheme (data from H. Barbas and C. Hilgetag, personal communication, overall neuron density data are given in Hilgetag et al. (2016), Table 4) were mapped to the FV91 scheme and partly estimated using the notion of architectural types. The translation uses the data of the most representative area in the original scheme. The classification of areas into architectural types (Hilgetag et al., 2016) extends a classification previously developed for prefrontal cortex (Barbas, 1986; Barbas & Rempel-Clower, 1997; Dombrowski et al., 2001) to the visual areas. Table 4 of Hilgetag et al. (2016) shows the classification which we translate to our area scheme as detailed in Table 6.4. The architectural type reflects the distinctiveness of the lamination as well as the degree of development of granular layer 4, with the lowest values corresponding to agranular and dysgranular cortices, and the highest value corre-

A: Model summary	
Populations	254 populations: 32 areas (Table 6.1) with eight populations each (area TH: six)
Topology	—
Connectivity	area- and population-specific but otherwise random
Neuron model	leaky integrate-and-fire (LIF), fixed absolute refractory period (voltage clamp)
Synapse model	exponential postsynaptic currents
Plasticity	—
Input	independent homogeneous Poisson spike trains
Measurements	spiking activity

B: Populations			
Type	**Elements**	**Number of populations**	**Population size**
Cortex	LIF neurons	32 areas with eight populations each (area TH: six), two per layer	N (area- and population-specific)

C: Connectivity	
Type	source and target neurons drawn randomly with replacement (allowing autapses and multapses) with area- and population-specific connection probabilities
Weights	fixed, drawn from normal distribution with mean w and standard deviation $\delta w = 0.1w$; 4E to 2/3E increased by factor 2 (cf. Potjans & Diesmann, 2014); weights of inhibitory connections increased by factor g; excitatory weights < 0 and inhibitory weights > 0 are redrawn, cortico-cortical weights onto excitatory and inhibitory populations increased by factor λ and $\lambda_{\mathcal{I}}\lambda$, respectively
Delays	fixed, drawn from Gaussian distribution with mean d and standard deviation $\delta d = 0.5d$; delays of inhibitory connections factor 2 smaller; delays rounded to the nearest multiple of the simulation step size $h = 0.1\,\mathrm{ms}$, inter-areal delays derived from distances with constant transmission speed; delays $< 0.1\,\mathrm{ms}$ before rounding are redrawn

D: Neuron and synapse model	
Name	LIF neuron
Type	leaky integrate-and-fire, exponential synaptic current inputs
Subthreshold dynamics	$\frac{\mathrm{d}V}{\mathrm{d}t} = -\frac{V-E_\mathrm{L}}{\tau_\mathrm{m}} + \frac{I_\mathrm{s}(t)}{C_\mathrm{m}}$ if $(\mathrm{t} > \mathrm{t}^* + \tau_\mathrm{r})$ $V(t) = V_r$ else $I_\mathrm{s}(t) = \sum_{i,k} w_k\, e^{-(t-t_i^k)/\tau_\mathrm{s}} \Theta(t - t_i^k)$ k: neuron index, i: spike index
Spiking	If $V(t-) < \theta \wedge V(t+) \geq \theta$ 1. set $t^* = t$, 2. emit spike with time stamp t^*

E: Input		
Type	**Target**	**Description**
Background	LIF neurons	independent Poisson spikes (see Table 6.3)

F: Measurements
Spiking activity

Table 3.1: Model description after Nordlie et al. (2009).

sponding to V1. Areas MIP and MDP were not classified due to insufficient data and were manually assigned type 5 like their neighboring area PO, which is similarly involved in visual reaching (Johnson et al., 1996; Galletti et al., 2003), and was placed at the same hierarchical level by Felleman & Van Essen (1991). To estimate neuron densities for areas not included in the data set, we compute the average density of layer v across areas with architectural type α.

2. Total cortical thicknesses $D(A)$ are given in Hilgetag et al. (2016) Table 4 for the same areas for which neuron densities were measured. These show a significant decrease with logarithmized overall neuron density (Fig. 3.1B). The total thicknesses of the remaining 18 areas are estimated using this linear fit. Quantitative data from the literature combined with our own estimates from a large number of published micrographs determine laminar thicknesses (Fig. 3.1C). For those areas for which no information is available, we estimate laminar thicknesses using a linear least-squares fit of per-area average relative thicknesses against logarithmized overall neuron density. Since the thicknesses of L2/3, L5, and L6 relative to the total thickness of cortex show no notable change with architectural type, we fill in missing values using the mean of the known data for these quantities. Relative L4 thickness increases with logarithmized total neuron density. For areas with missing data these linear regression fits provide relative L4 thicknesses. Absolute thicknesses then follow from the product of relative thickness and total thickness.

3. The fraction $\gamma(v)$ of excitatory neurons in layer v is assumed to be identical across areas. For the laminar dependency, the values that Binzegger et al. (2004) observe in cat V1 are used (see Table 3.2).

layer	$\frac{N_\mathrm{E}}{N_\mathrm{E}+N_\mathrm{I}}$	$\frac{N_\mathrm{I}}{N_\mathrm{E}+N_\mathrm{I}}$
2/3	0.78	0.22
4	0.80	0.20
5	0.82	0.18
6	0.83	0.17

Table 3.2: Proportions of inhibitory and excitatory neurons in each layer of cat primary visual cortex according to Binzegger et al. (2004).

These data and assumptions deliver the number of neurons in population i of area A,

$$N(A,i) = \rho\left(A, v_i\right) S(A) D\left(A, v_i\right) \cdot \begin{cases} \gamma\left(v_i\right) & \text{if } i \in \mathcal{E} \\ 1-\gamma\left(v_i\right) & \text{if } i \in \mathcal{I} \end{cases}, \tag{3.1}$$

where v_i denotes the layer of population i, $S(A)$ the surface area of area A computed with the Caret software (Van Essen et al., 2001) on the basis of its representation on the F99 cortical surface (Van Essen, 2002), $D(A, v_i)$ the thickness of layer v_i, and $\mathcal{E}, \mathcal{I}$ the pool of excitatory and inhibitory populations, respectively. Table 6.7 lists the surface areas $S(A)$ we use in the derivation of the connectivity.

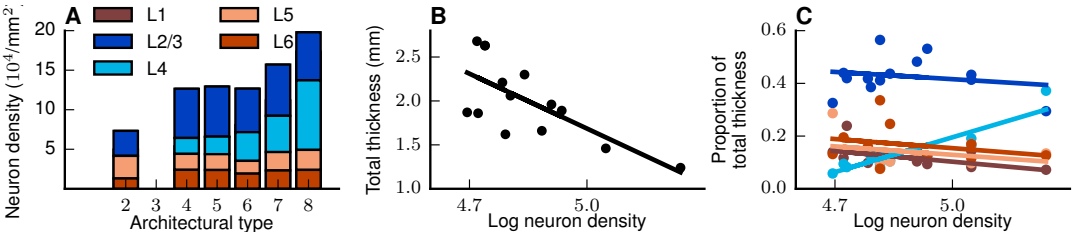

Figure 3.1: Cortical architecture. (A) Laminar neuron densities for the architectural types in the model. Type 2, here corresponding only to area TH, lacks L4. In the model, L1 contains synapses but no neurons. Data provided by H. Barbas and C. Hilgetag (personal communication) and linearly scaled up to account for undersampling of cells by NeuN staining relative to Nissl staining as determined by repeat measurements of 11 areas. (B) Total thickness vs. logarithmized overall neuron density and linear least-squares fit ($r = -0.7$, $p = 0.005$). (C) Relative laminar thickness vs. logarithmized overall neuron density and linear least-squares fits (L1: $r = -0.51$, $p = 0.08$, L2/3: $r = -0.20$, $p = 0.52$, L4: $r = 0.89$, $p = 0.0001$; L5: $r = -0.31$, $p = 0.36$, L6: $r = -0.26$, $p = 0.43$). Total cortical thicknesses D(A) and overall neuron densities for 14 areas provided by H. Barbas and C. Hilgetag (personal communication), measured by Nissl staining for the 11 areas mentioned above and for 3 areas by NeuN staining and linearly scaled up to account for undersampling.

Each neuron in the network receives synapses of four different origins, as sketched in Fig. 3.2A. In the following, we describe how the counts for each of these synapse types are computed.

3.3 Local connectivity

In line with a number of anatomical observations (Harrison et al., 2002; O'Kusky & Colonnier, 1982; Schüz & Palm, 1989; Cragg, 1967), we assume that the total synaptic volume density is constant across areas. Cragg (1967) provides an average indegree in monkey visual cortex of $\sim 5,600$ synapses per neuron while O'Kusky & Colonnier (1982) report $2,300$ synapses per neuron. We take the average $3,950$ of both values as representative for V1. The resulting volume density is $\rho_{\text{syn}} = \frac{3950\, N(V1)}{D(V1)\, S(V1)} = 5.9 \cdot 10^8 \frac{\text{synapses}}{\text{mm}^3}$, which approximately agrees with $6.3 \cdot 10^8 \frac{\text{synapses}}{\text{mm}^3}$ reported for somatosensory cortex of rat by Markram et al. (2015).

The connection probabilities of the microcircuit model listed in Table 5 of Potjans & Diesmann (2014), which have been computed from various anatomical and electrophysiological studies (with large contributions from Binzegger et al., 2004; Thomson & Lamy, 2007), serve as the basis of our calculations. The connectivity between any pair of populations in the model is spatially uniform. However, to derive the corresponding connection probabilities,

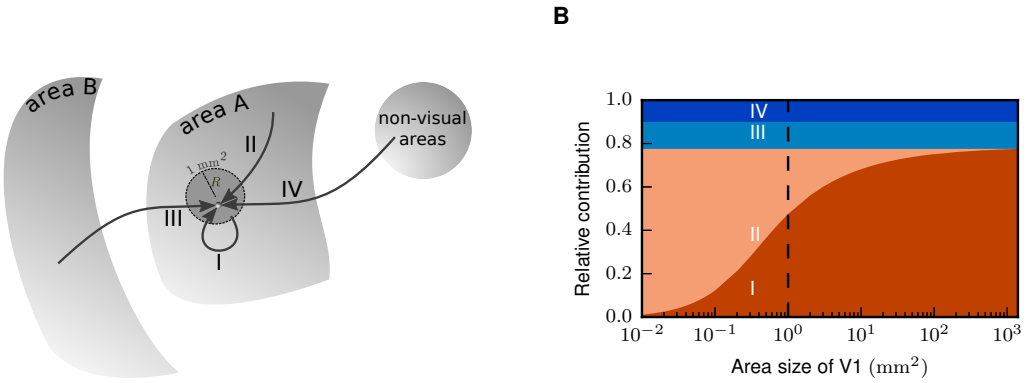

Figure 3.2: Contributions of different synaptic sources. A Scheme of the different types of connections to a cortical neuron. I: Simulated intra-areal synapses, II: Intra-areal synapses from outside the $1\ \mathrm{mm}^2$ patch modeled as Poisson sources, III: Simulated cortico-cortical synapses, IV: Synapses from subcortical and non-visual cortical areas modeled as Poisson sources. **B** Relative contributions to indegrees in V1 (stacked vertically) for increasing cortical surface area (horizontal) covered by the model. The number of simulated intra-areal synapses (red, I) increases at the cost of random input from non-simulated intra-areal synapses (light red, II) while the number of cortico-cortical synapses (light blue, III) and random inputs from subcortical and non-visual cortical areas (blue, IV) stays constant. The vertical dashed line signals the surface area of $1\ \mathrm{mm}^2$ used in the present study.

we approximate the underlying probability C for a given pair of neurons to establish at least one contact as falling off with distance according to a Gaussian probability distribution with standard deviation $\sigma = 297\,\mu\mathrm{m}$ (Potjans & Diesmann, 2014). To determine average connection probabilities, we approximate each brain area A as a flat disk with (area-specific) radius R and assign polar coordinates r and θ to each neuron. The radius R hence determines the cut-off of the Gaussian distribution and the precise values of the population connectivities. The average connection probability between a pre- and postsynaptic neuron pair is then obtained by integrating over all possible positions of the two neurons:

$$\bar{C}(R) = \frac{C_0}{\pi^2 R^4} \int_0^R \int_0^{2\pi} \int_0^R \int_0^{2\pi} \exp\left[\frac{-\left(r_1^2 + r_2^2 - 2r_1 r_2 \cos(\theta_1 - \theta_2)\right)}{2\sigma^2}\right] r_1 r_2 \mathrm{d}\theta_1 \mathrm{d}r_1 \mathrm{d}\theta_2 \mathrm{d}r_2\,, \quad (3.2)$$

where C_0 is the connection probability at zero distance. This can be reduced to a simpler form (Sheng, 1985), reading

$$\bar{C}(R) = \frac{2C_0}{\pi R^2} \int_0^{2R} e^{-r^2/2\sigma^2} \left[4 \arctan\left(\frac{2R - r}{2R + r}\right)^{1/2} - \sin\left(4 \arctan\left[\frac{2R - r}{2R + r}\right]^{1/2}\right)\right] r \mathrm{d}r\,. \quad (3.3)$$

Averaged across pairs of populations, the value of C_0 is 0.143 (computed from Eq. 8 and Table S1 in Potjans & Diesmann, 2014)). Note that Potjans & Diesmann (2014) use a simpler approach where only the position of one neuron is varied while keeping the other neuron fixed in the

center of the disk (Eq. 9 in that paper). Connection probabilities computed with the latter approach are denoted with the subscript PD14. Moreover, in the following, we use primes for all variables referring to a network with the laminar distribution of neurons of the microcircuit model.

We use Eq. 3.3 to determine the population-specific spatially averaged connection probabilities. The parameters of the microcircuit model are reported for a 1 mm^2 patch of cortex, corresponding to $R = \sqrt{1/\pi}$ mm, which we call R_0. For each source population j and target population i, we first translate the connection probabilities of the 1 mm^2 model to area-dependent R via

$$C'_{ij}(R) = C'_{ij,\mathrm{PD14}}\left(R_0\right) \frac{\bar{C}'(R)}{\bar{C}'_{\mathrm{PD14}}\left(R_0\right)} ,$$

with $\bar{C}'_{\mathrm{PD14}}(R_0) = 0.066$. From the connection probability, we compute the number of synapses for a projection from population j to i as

$$N_{\mathrm{syn},ij} = \frac{\log\left(1 - C_{ij}\right)}{\log\left(1 - \frac{1}{N_i N_j}\right)} ,$$

which follows from randomly drawing source and target neurons with replacement (cf. Eq. 1 in Potjans & Diesmann, 2014). The indegree K_{ij} is the number of incoming synapses per target neuron, $N_{\mathrm{syn},ij}/N_i$. In the following, all numbers of synapses $N_{\mathrm{syn}}(A)$ and indegrees $K_{ij}(A)$ are area-specific. For simplicity, we leave out the argument A. Since mean synaptic inputs are proportional to the indegrees, we consider them as a defining characteristic of the local circuit and preserve their relative values to approximately preserve relative influences between populations when adjusting the model to area-specific neuron densities,

$$\begin{aligned} \frac{K_{ij}(R)}{K_{kl}(R)} &= \frac{K'_{ij}(R)}{K'_{kl}(R)} \;\; \forall i,j,k,l \\ \Leftrightarrow K_{ij}(R) &= c_A(R) K'_{ij}(R) \;\; \forall i,j \,, \end{aligned} \tag{3.4}$$

where $c_A(R)$ is an area-specific conversion factor, which is larger for areas with smaller neuron densities because of the assumption of a constant synaptic volume density. The total number of synapses local to the patch (type I) is the sum over the projections between all populations of the area:

$$N_{\mathrm{syn,I}}(R) = \sum_{i,j} N_i(R) K_{ij}(R) = c_A(R) \sum_{i,j} N_i(R) K'_{ij}(R) \,.$$

We thus obtain $c_A(R)$ by determining $N_{\mathrm{syn,I}}$. To this end, we use retrograde tracing data from

Markov et al. (2011) consisting of fractions of labeled neurons (FLN) per area as a result of injections into one area at a time. The fraction intrinsic to the injected area, FLN_i, is approximately equal for all 9 areas where this fraction was determined, with a mean of 0.79. We translate this into numbers of synapses by assuming that, on average, each labeled neuron establishes the same number of synapses in the injected area, so that the proportion of synapses of type I is 0.79 for the full area size. For areas modeled with reduced size, this fraction is smaller because, in that case, synapses of both type I and II contribute to the value of 0.79 (Fig. 3.2B). We approximate the increasing contribution of type I synapses with the modeled area size as the increase in indegrees averaged over population pairs,

$$\frac{N_{\mathrm{syn,I}}(R)/N_{\mathrm{syn,tot}}(R)}{N_{\mathrm{syn,I}}(R_\mathrm{full})/N_{\mathrm{syn,tot}}(R_\mathrm{full})} = \left\langle \frac{K_{ij}(R)}{K_{ij}(R_\mathrm{full})} \right\rangle_{ij} = \left\langle \frac{K'_{ij}(R)}{K'_{ij}(R_\mathrm{full})} \right\rangle_{ij},$$

where in the last step we use Eq. 3.4. Using $N_{\mathrm{syn,I}}(R_\mathrm{full})/N_{\mathrm{syn,tot}}(R_\mathrm{full}) = FLN_\mathrm{i}$, we obtain

$$N_{\mathrm{syn,I}}(R) = N_{\mathrm{syn,tot}}(R)\, FLN_\mathrm{i} \left\langle \frac{K'_{ij}(R)}{K'_{ij}(R_\mathrm{full})} \right\rangle_{ij},$$

where $N_{\mathrm{syn,tot}}(R) = \rho_\mathrm{syn}\pi R^2 D$ with D the total thickness of the given area. The conversion factor can thus be obtained with

$$c_A(R) = \frac{N_{\mathrm{syn,tot}}(R)}{\sum_{i,j} N_i(R) K'_{ij}(R)}\, FLN_\mathrm{i} \left\langle \frac{K'_{ij}(R)}{K'_{ij}(R_\mathrm{full})} \right\rangle_{ij}.$$

We substitute this into Eq. 3.4 for the modeled areas where $R = R_0$ and obtain the population-specific indegrees for synapses of type I:

$$K_{ij,\mathrm{I}} := K_{ij}\,(R = R_0) \tag{3.5}$$

3.4 Cortico-cortical connectivity

In the model, all synapses onto a neuron in the target area originate exclusively in the 1 mm^2 patch representing the source area. This is similar to assuming that the connections between areas are parallel on the 1 mm^2 scale. In nature, these connections exhibit a certain degree of spatial convergence and divergence (Colby et al., 1988; Salin et al., 1989; Gattass et al., 1997; Markov et al., 2014b). However, our choice enhances the interactions between areas and counteracts the dilution of inter-areal influences due to the lack of spatial organization within the modeled patch.

As a starting point, we determine whether a pair of areas is connected using the union

of all connections available in the FV91 scheme in the CoCoMac database (Stephan et al., 2001; Bakker et al., 2012; Suzuki & Amaral, 1994a; Felleman & Van Essen, 1991; Rockland & Pandya, 1979; Barnes & Pandya, 1992) (Fig. 3.3A, see Sec. 6.4.9 for details) and all connections reported by Markov et al. (2014a). We use a new release of CoCoMac, in which mappings from brain regions in other nomenclatures were scrutinized to ensure a consistent transfer of connections into the FV91 name space. We then determine the population-specific numbers of modeled cortico-cortical synapses in three steps: 1. determining the area-level connectivity; 2. distributing synapses across layers; 3. assigning synapses to target neurons.

For the first step, we compute the total numbers of synapses formed between each pair of areas using retrograde tracing data from Markov et al. (2014a). The data consist of fractions of labeled neurons $FLN_{AB} = NLN_{AB} / \sum_{B'} NLN_{AB'}$, with NLN the number of labeled neurons in area B following an injection in area A. Markov et al. (2014a) used a parcellation scheme called M132 which is also available as a cortical surface, both in native and in F99 space. On the target side we use the coordinates of the injection sites registered to the F99 atlas available via the Scalable Brain Atlas (Bakker et al., 2015) to identify the equivalent area in the FV91 parcellation (cf. Table 6.9). There is data for 11 visual areas in the FV91 scheme with repeat injections in six areas, for which we take the arithmetic mean. To map data on the source side from the M132 atlas to the FV91 parcellation, we count the number of overlapping triangles on the F99 surface between any given pair of regions and distribute the FLN proportionally to the amount of overlap, using tools available at the CoCoMac site (http://cocomac.g-node.org/services/f99_region_overlap.php). Fig. 3.3B shows the result of this mapping procedure. To estimate values for the areas not included in the data set, we make use of a roughly exponential fall-off of connectivities with distance (Ercsey-Ravasz et al., 2013), modeled as

$$FLN_{AB} = C \cdot \exp\left(-\lambda d_{AB}\right) . \tag{3.6}$$

A linear least-squares fit of the logarithm of the FLN (Fig. 3.3C) predicts missing values. The total number of synapses $N_{\mathrm{syn},AB}$ between each pair of areas $\{A, B\}$ is assumed to be proportional to the number of labeled neurons NLN_{AB} and thus to FLN_{AB},

$$\frac{N_{\mathrm{syn},AB}}{\underbrace{\sum_{B'} N_{\mathrm{syn},AB'}}_{=N_{\mathrm{syn,tot},A}}} = \frac{NLN_{AB}}{\sum_{B'} NLN_{AB'}} = \frac{FLN_{AB}}{\sum_{B'} FLN_{AB'}}. \tag{3.7}$$

This corresponds to individual neurons in each source area (including area A itself) on average establishing the same number of synapses in the target area A. For each target area, the FLN in the model should add up to the total fraction of connections coming from visual cortical

areas. However, this fraction is not known a priori. To provide a normalization, we therefore consider also non-visual areas, for which distances are available and for which we can hence also estimate the FLN. The total fraction of all connections from subcortical regions was found to average 1.3 % in eight cortical areas (Markov et al., 2011). This allows us to normalize the combined FLN from all cortical areas as $\sum_B FLN_{AB} = 1 - FLN_{\mathrm{i}} - 0.013$, where B here runs over both modeled and non-modeled cortical areas. Fig. 3.3D shows the result of combining the binary information from CoCoMac with measured and estimated FLN to a connectivity matrix on the area level.

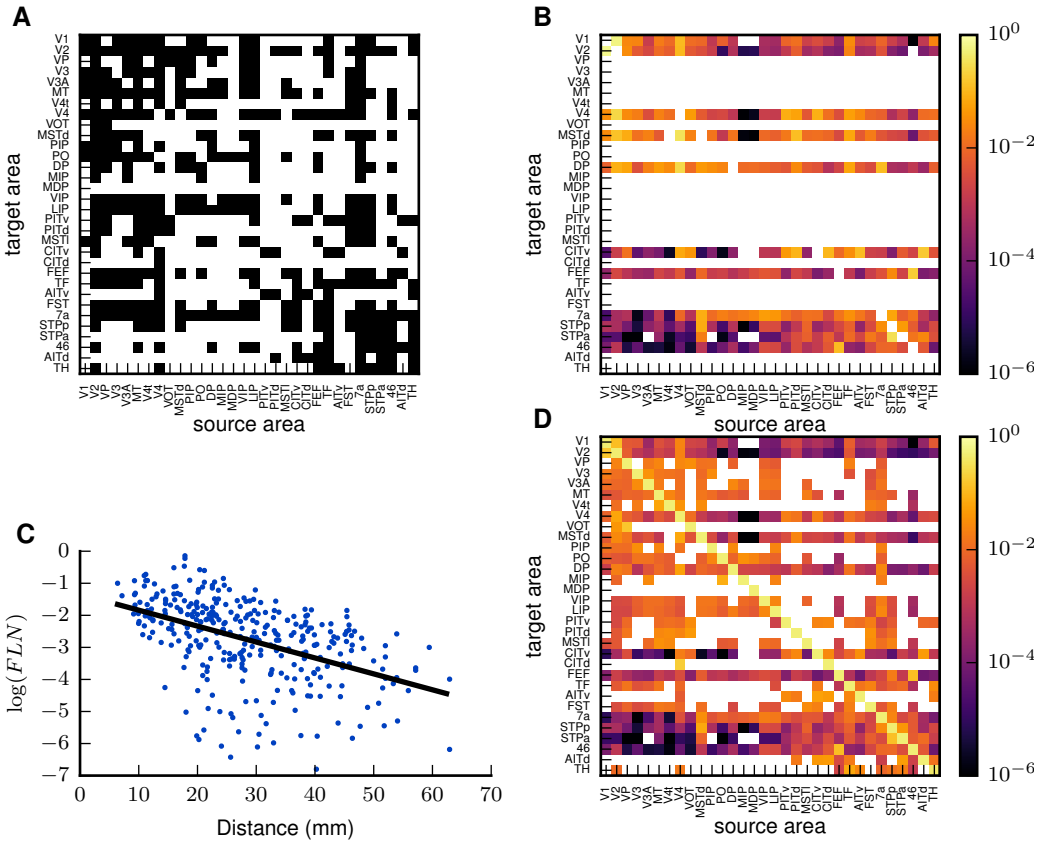

Figure 3.3: Construction of the connectivity map at area level. **A** Binary connectivity information from the CoCoMac database. Black, existing connections; white, absent connections. **B** Fractions of labeled neurons (FLN) from Markov et al. (2014a) mapped to the parcellation scheme of Felleman & Van Essen (1991). **C** Connection densities fall off approximately exponentially with inter-area distance. Distances computed as median vertex distance in the cortical surface representation of the M132 areas. Black line: linear regression with $\log(FLN) = \log(C) - \lambda d$ ($C = 0.045, \lambda = 0.11, p = 10^{-19}$; cf. Eq. 3.6). **D** Area-level connectivity of the model, based on data shown in panels A-C, expressed as relative indegrees for each target area.

As a next step, we determine the distribution of connections across source and target

layers. Cortico-cortical connections tend to follow specific laminar patterns that are related to the differences in architectural types or neuron densities of the areas (Hilgetag et al., 2016), and consequently also to their relative positions in an approximate processing hierarchy (Felleman & Van Essen, 1991). On the source side, the laminar pattern of projections can be expressed as the fraction of supragranular labeled neurons (SLN) in retrograde tracing experiments (Markov et al., 2014b). To determine the SLN entering into the model, we follow a similar procedure as for the FLN: We use the exact coordinates of the injections to determine the corresponding target area A in the FV91 parcellation, and in case of repeat injections into the given target area, for each pair of areas we take the mean SLN across injections. To map the data from the M132 parcellation to the FV91 scheme, we weight the SLN by the overlap $c_{B,\beta}$ between area β in the former and area B in the latter scheme and the FLN to take into account the overall strength of the connection, so that

$$SLN_{AB} = \frac{\sum_\beta c_{B,\beta} FLN_{A,\beta} SLN_{A,\beta}}{\sum_\beta c_{B,\beta} FLN_{A,\beta}} \,.$$

Since SLN data are not available for all pairs of areas, we estimate missing values using a sigmoidal relation between SLN and the logarithmized ratio of overall cell densities of the two areas (Fig. 3.4A). This is similar to the relation between SLN and hierarchical distances found by Markov et al. (2014b), but avoids the cyclicity of estimating SLN from the hierarchy which itself was obtained using the SLN data. A relationship between laminar patterns and log ratios of neuron densities was suggested by Beul et al. (2015). Following Markov et al. (2014b), we use a beta-binomial generalized linear model (GLM), which assumes the numbers of labeled neurons in the source areas to sample from a beta-binomial distribution (e.g. Weisstein, 2005). This distribution arises as a combination of a binomial distribution with parameter p giving the probability of supragranular labeling within each source area, and a beta distribution of p across areas with dispersion parameter ϕ. With the probit link function g (e.g. McCulloch et al., 2008), the relation between the measured SLN and the log ratio ℓ of neuron densities for each pair of areas becomes

$$g(SLN) = a_0 \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + a_1 \ell, \tag{3.8}$$

where ℓ and SLN are vectors and $\{a_0, a_1\}$ are scalar fit parameters. To fit SLN to logarithmized ratios of cell densities, we map the FV91 areas to the Markov et al. (2014b) scheme with the overlap tool of CoCoMac (see above) and compute the cell density of each area in the M132 scheme as a weighted average over the relevant FV91 areas (each with cell density either mea-

sured, or estimated according to its architectural type). For areas with identical names in both schemes, we simply take the neuron density as determined in the FV91 scheme. Fig. 3.4A shows the result of the SLN fit carried out in R (R Core Team, 2015) with the betabin function of the aod package (Lesnoff & Lancelot, 2012). In contrast to Markov et al. (2014b), who excluded certain areas when fitting SLN vs. hierarchical distances in view of ambiguous hierarchical relations, we take all data points into account to obtain a simple and uniform rule.

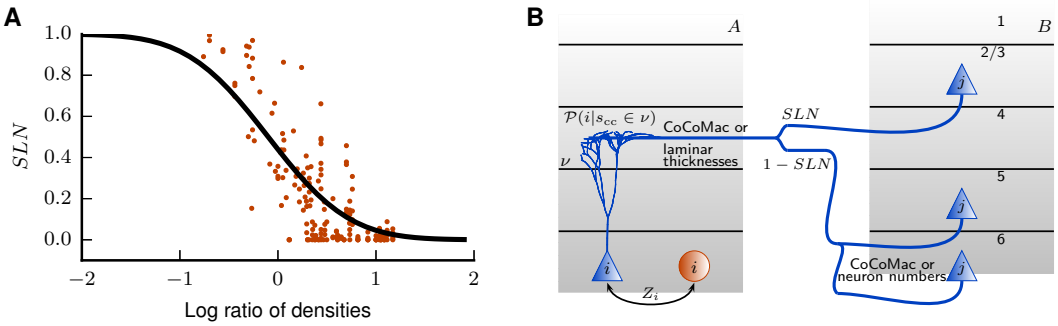

Figure 3.4: Laminar patterns of cortico-cortical connections. A Fractions of source neurons in supragranular layers depend sigmoidally on the logarithmized ratio of the overall cell densities of the two areas. SLN data from Markov et al. (2014b), neuron densities from Hilgetag et al. (2016). Black curve: fit using a beta-binomial generalized linear model Eq. 3.8 ($a_0 = -0.152$, $a_1 = -1.534$, $\phi = 0.214$). **B** Schematic illustration of the procedure for distributing synapses across layers. Source neuron j from area B sends an axon to layer v of area A where a cortico-cortical synapse s_{CC} is formed at the dendrite of neuron i. The inclusion of a layer in a source or target pattern is determined by CoCoMac data if available, and otherwise based on measured or estimated SLN. The diagram illustrates the distribution of synapses for connected sets of layers. To distribute synapses across excitatory populations in the sending area, SLN is combined with projection densities from CoCoMac if available, or with neuron numbers as proportionality factors. Layer 1 is only modeled as a target layer where synapses are formed and associated with cell bodies in layers $2-6$. In the target area, synapses are distributed across the layers contained in the target pattern using projection densities from CoCoMac if available, or in proportion to laminar thicknesses. The synapses are then attributed to neurons with the help of data from Binzegger et al. (2004). Finally, synapses are redistributed between excitatory and inhibitory neurons such that $95\,\%$ of synapses target excitatory neurons. See Eq. 3.15 for the formal definitions.

As a further step toward determining laminar distributions of synapses, we combine the measured or estimated SLN with data from CoCoMac. The SLN data and CoCoMac complement each other, with the SLN being quantitatively more precise and CoCoMac containing more detailed laminar patterns. Specifics of the processing of the laminar CoCoMac data are given in Sec. 6.4.9. CoCoMac gives the connection strength $\alpha(v)$ of a layer v on a rough quantitative scale, ranging from 0 (absent) to 3 (strong) in integer steps. To distribute synapses between layers, we assume that the connection strengths from CoCoMac represent numbers of synapses in orders of magnitude. The inclusion of a layer in the source pattern P_{s} is based on CoCoMac (Felleman & Van Essen, 1991; Barnes & Pandya, 1992; Suzuki & Amaral, 1994b; Morel & Bullier, 1990; Perkel et al., 1986; Seltzer & Pandya, 1994) if the corresponding data is available (this is the case for $45\,\%$ of all cortico-cortical connections); otherwise, we include

layers 2/3, 5 and 6 in the calculation. The exclusion of granular layer 4 from the source pattern is in line with most anatomical observations (Felleman & Van Essen, 1991). Furthermore, we model cortico-cortical connections as emanating from excitatory neurons only, which is a good approximation to experimental findings (Salin & Bullier, 1995; Tomioka & Rockland, 2007). If a layer is included in the source pattern, we assign a fraction of the total outgoing synapses to it according to the SLN. Since the SLN does not further distinguish between the infragranular layers 5 and 6, we use the rough connection densities from CoCoMac for this purpose when available, and otherwise we assume a constant outdegree, distributing synapses in proportion to the numbers of neurons. On the target side, we determine the pattern of target layers P_t from anterograde tracer studies contained in CoCoMac (Jones et al., 1978; Rockland & Pandya, 1979; Morel & Bullier, 1990; Webster et al., 1991; Felleman & Van Essen, 1991; Barnes & Pandya, 1992; Distler et al., 1993; Suzuki & Amaral, 1994b; Webster et al., 1994), which are available for 29 % of all cortico-cortical connections. For connections without laminar information on the target side, we use termination patterns suggested by the SLN based on a relationship between source and target patterns. Using the terminology of visual hierarchies, we denote projections with low, intermediate, and high SLN respectively as feedback, lateral, and feedforward projections. A finer classification may be obtained by simultaneously considering target patterns (Felleman & Van Essen, 1991, Figure 3). We take $SLN < 0.35$ to correspond to feedback projections, $SLN > 0.65$ to feedforward projections and $SLN \in [0.35, 0.65]$ to lateral projections. Feedforward projections terminate preferentially in layer 4 (F), feedback projections tend to have a multilaminar (M) termination pattern, and lateral projections tend to have a columnar (C) termination pattern encompassing all layers (Felleman & Van Essen, 1991). In this case, the termination pattern P_t can thus be of the following types:

$$\begin{aligned} F =& \{4\} \text{ for } SLN > 0.65 \\ M =& \{1, 2/3, 5, 6\} \text{ for } SLN < 0.35 \\ C =& \{1, 2/3, 4, 5, 6\} \text{ for } SLN \in [0.35, 0.65] \end{aligned} \quad , \tag{3.9}$$

and we distribute synapses among the layers in the termination pattern in proportion to their thickness.

Since we use a point neuron model, we have to account for the possibly different laminar positions of cell bodies and synapses. The data of Binzegger et al. (2004) deliver three quantities that allow us to relate synapse to cell body locations. The first is the probability $\mathcal{P}(s_\mathrm{cc}|c_\mathrm{B} \bigcap s \in v)$ for a synapse in layer v on a cell of type c_B (e.g., a pyramidal cell with soma in layer 5) to be of cortico-cortical origin. Second, Binzegger et al. (2004) provide the relative occurrence

$\mathcal{P}(c_{\mathrm{B}})$ of the cell type c_{B}, and third, the total numbers of synapses $N_{\mathrm{syn}}(v, c_{\mathrm{B}})$ in layer v onto the given cell type. We map these data to our model by computing the conditional probability $\mathcal{P}(i|s_{\mathrm{cc}} \in v)$ for the target neuron to belong to population i if a cortico-cortical synapse s_{cc} is located in layer v. This probability equals the sum of probabilities that a synapse is established on the different Binzegger subpopulations making up our populations,

$$\mathcal{P}(i|s_{\mathrm{cc}} \in v) = \mathcal{P}(\bigcup_{c_{\mathrm{B}} \in i} c_{\mathrm{B}}|s_{\mathrm{cc}} \in v) = \sum_{c_{\mathrm{B}} \in i} \mathcal{P}(c_{\mathrm{B}}|s_{\mathrm{cc}} \in v)\,. \tag{3.10}$$

where

$$\mathcal{P}(c_{\mathrm{B}}|s_{\mathrm{cc}} \in v) = \frac{\mathcal{P}(c_{\mathrm{B}} \bigcap s_{\mathrm{cc}} \in v)}{\mathcal{P}(s_{\mathrm{cc}} \in v)}\,. \tag{3.11}$$

The numerator gives the joint probability that a cortico-cortical synapse is formed in layer v on cell type c_{B},

$$\mathcal{P}(c_{\mathrm{B}} \bigcap s_{\mathrm{cc}} \in v) = \frac{N_{\mathrm{syn,CC}}(v, c_{\mathrm{B}})\mathcal{P}(c_{\mathrm{B}})}{\sum_{v', c'_{\mathrm{B}}} N_{\mathrm{syn,CC}}(v', c'_{\mathrm{B}})\mathcal{P}(c'_{\mathrm{B}})}\,, \tag{3.12}$$

and the denominator equals the probability of finding a cortico-cortical synapse in layer v, which is computed by summing over cell types,

$$\mathcal{P}(s_{\mathrm{cc}} \in v) = \sum_{c_{\mathrm{B}}} \mathcal{P}(c_{\mathrm{B}} \bigcap s_{\mathrm{cc}} \in v)\,. \tag{3.13}$$

$N_{\mathrm{syn,CC}}(v, c_{\mathrm{B}})$ represents the number of cortico-cortical synapses in layer v on cell type c_{B},

$$N_{\mathrm{syn,CC}}(v, c_{\mathrm{B}}) = \mathcal{P}(s_{\mathrm{cc}}|c_{\mathrm{B}} \bigcap s \in v)\, N_{\mathrm{syn}}(v, c_{\mathrm{B}})\, \mathcal{P}(c_{\mathrm{B}})\,, \tag{3.14}$$

which can be directly determined from the data.

Combining these equations, we obtain the number of cortico-cortical (Type III) synapses from excitatory population j of area B to population i of area A (cf. Fig. 3.4B):

$$N_{\text{syn,III}}(i,A,j,B) = \underbrace{Z_i \sum_{v \in P_\text{t}} Y_v \, \mathcal{P}(i|s_\text{cc} \in v)}_{\text{target side}} \underbrace{X_j}_{\text{source side}} N_{\text{syn,III}}(A,B)\,, \tag{3.15}$$

$$\text{with } X_j = \begin{cases} SLN & \text{if } j \in S \bigcap P_\text{s} \\ (1-SLN)\frac{10^{\alpha(v_j)}}{\sum_{j' \in I,\, \alpha(v_{j'})>0} 10^{\alpha(v_{j'})}} & \text{if } j \in I \text{ and } \alpha(v_j) > 0 \\ (1-SLN)\frac{N(A,j)}{\sum_{j' \in I} N(A,j')} & \text{if } j \in I \bigcap P_\text{s} \text{ but no CoCoMac data available} \\ 0 & \text{if } j \notin P_\text{s} \end{cases},$$

$$\text{and } Y_v = \begin{cases} \frac{10^{\alpha(v)}}{\sum_{\alpha(v')>0} 10^{\alpha(v')}} & \text{if } \alpha(v) > 0 \\ \frac{D(A,v)}{\sum_{v'} D(A,v')} & \text{if no CoCoMac data available} \end{cases}.$$

Here, $S = 2/3\text{E}$ and $I = \{5E, 6E\}$ respectively denote the supragranular and infragranular excitatory populations. Z_i is an additional factor which takes into account the experimentally found phenomenon of E-I-specificity in feedback connections. Various studies have found that cortico-cortical feedback connections preferentially target excitatory neurons rather than inhibitory neurons (Johnson & Burkhalter, 1996, 1997; Anderson et al., 2011). Z_i is area-specific and depends on the excitatory or inhibitory nature of the target population, but not on the target layer. Different values for the proportion of excitatory targets have been found, ranging between 87 % and 98 %. As a representative value, we choose a fraction of 93 % of connections targeting excitatory population in the target area. For each feedback connection in the model, we thus redistribute the synapses across the excitatory and inhibitory target populations and determine Z_i such that

$$\frac{\sum_{i \in \mathcal{E}} \sum_j N_{\text{syn,III}}(i,A,j,B)}{N_{\text{syn,III}}(A,B)} = 0.93\,.$$

Fig. 6.2 shows the resulting connection probabilities between all pairs of populations in the model.

3.5 External, random input

Finally, inputs at synapses formed outside the scope of our model, i.e., white-matter synapses from non-cortical or non-visual cortical areas as well as gray-matter synapses formed by cells outside the $1\,\text{mm}^2$ patch, are replaced by Poisson spike trains. The external inputs to each population of the microcircuit model of Potjans & Diesmann (2014) are estimated as a sum over

gray- and white-matter inputs and thalamic afferents, and are representative of V1. However, corresponding experimental data are not consistently available across the areas of our model. Furthermore, since laminar patterns of external inputs from cortical areas (Markov et al., 2014b) as well as from subcortical regions such as pulvinar (Felleman & Van Essen, 1991; Rockland et al., 1999) differ between areas, we cannot simply copy the numbers for V1 to the remaining areas. In the absence of area-specific data, we use a simple scheme: For each area, we compute the total number of external synapses as the difference between the total number of synapses and those of type I and III and distribute these such that all neurons in the given area have the same indegree for Poisson sources. The only exception is that in area TH, we compensate for the missing granular layer 4 by increasing the external drive onto populations 2/3E and 5E by 20 %.

3.6 Conclusion

We have presented the construction of a multi-area model of macaque visual cortex in a modular and algorithmic manner facilitating further refinements as new data become available. The resulting formal network specification enables simulations of the spiking dynamics using the NEST simulator (Gewaltig & Diesmann, 2007). In the next chapter, we will present the analytical theory used to refine the model definition by constraining it with experimental findings on cortical dynamics.

CHAPTER 4

Integration of dynamical constraints by analytical theory

The following chapter has been published in Schuecker J, Schmidt M, van Albada SJ, Diesmann M, and Helias M (2015). Fundamental activity constraints lead to specific interpretations of the connectome. arXiv preprint arXiv:1509.03162 (Schuecker et al., 2015b). The conclusion is an excerpt from the discussion of the paper.

Preliminary results have been presented in Schmidt et al. (2015b).

Author contributions: Jannis Schücker and the author performed the work concertedly with equal shares on all parts of the study. Sacha J. van Albada, Markus Diesmann and Moritz Helias supervised the work and contributed to the writing of the manuscript.

4.1 Motivation

The neural wiring diagram, the connectome, is gradually being filled through classical techniques combined with innovative quantitative analyses (Markov et al., 2014a,b) and new technologies (Wedeen et al., 2008; Axer et al., 2011). The connectivity between neurons is considered to shape resting-state and task-related collective activity (Cole et al., 2014; Shen et al., 2015). For simple networks, clear relationships with activity are known, even analytically, e.g., a dynamic balance between excitatory and inhibitory inputs in inhibition-dominated random networks leads to an asynchronous and irregular state (van Vreeswijk & Sompolinsky, 1996; Amit & Brunel, 1997a). Structures and mechanisms underlying large-scale interactions have been identified by means of dynamical models (Deco & Jirsa, 2012; Cabral et al., 2014) and graph theory (Markov et al., 2013; Goulas et al., 2014). Furthermore, the impact of local network structure on spike-time correlations is known in some detail (Ostojic et al., 2009; Pernice et al., 2011; Trousdale et al., 2012). Under certain conditions, there is a one-to-one mapping between correlations in neuronal network activity and effective connectivity, a measure that depends on the network structure and on its activity (van Albada et al., 2015). In conclusion, anatomy does not uniquely determine dynamics, but dynamical observations help constrain the underlying anatomy. Despite advances in the understanding of simple networks, a complete picture of the relationship between structure and dynamics remains elusive.

Fig. 4.1 visualizes the integrative loop between experiment, model, and theory that guides the investigation of structure and dynamics. In the traditional forward modeling approach, structural data from experimental studies determines the connectivity between model neurons. Combined with the specification of a particular single-neuron dynamics and synaptic interactions, simulations yield a certain network dynamics. There is a fundamental problem with this approach.

Despite the impressive experimental progress in determining network parameters, any neural network model is underdetermined, because of the sheer complexity of the brain tissue and inevitable uncertainties in the data. For instance, counting synapses on a large scale presently takes a prohibitive amount of time, and no available technique allows determining precise synaptic weights for entire populations of neurons. Although it may be possible to quantify the full connectome of an individual in future, inter-individual variability will require modelers to express connectivity as connection probabilities or rules of self-organization if they want to learn about general principles of the brain. In modeling studies, parameters are usually tuned manually to achieve a satisfactory state of activity, which becomes unfeasible for high-dimensional models due to the size of the parameter space.

In particular, it is a priori unclear how parameters of the model influence its activity. In

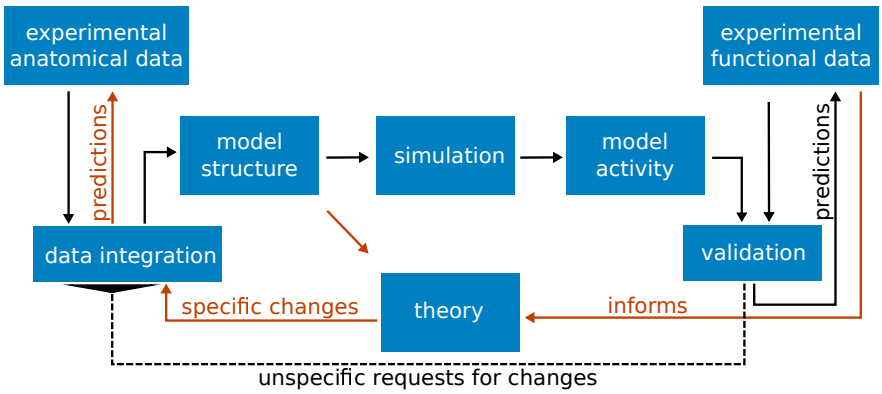

Figure 4.1: The integrative loop between experiment, model, and theory. Black arrows represent the classical forward modeling approach: Experimental structural data are integrated into a model structure, which gives rise to the activity via simulation. The model activity is compared to experimental functional data. The usual case of disagreement leads to the need to change the model definition. By experience and intuition the researcher identifies the parameters to be changed, proceeding in an iterative fashion. Once the model agrees well with a number of experimental findings, it may also generate predictions on the outcome of new experiments. Red arrows symbolize our new approach: informed by functional data, an analytical theory of the model identifies critical connections used to refine the integration of data into a consistent model structure and to generate predictions for anatomical studies.

consequence, modifications cannot be performed in a targeted fashion, and it is difficult to find a minimal set of modifications necessary for aligning a model with the experimental activity data.

Overcoming this problem requires a shift of perspective. Instead of regarding the model exclusively in a forward manner, generating predicted activity from structure, we in addition consider the system in a reverse manner, predicting the structure necessary to explain the observed activity. Our theory, starting from a mean-field description, provides a direct link between structure and activity. In contrast to simulations, the theory is invertible, which we exploit to identify connections critical for the dynamics and to find a minimal set of modifications to the structure yielding a realistic set point of activity. The predicted alterations of the structure generate hypotheses on brain structure, thus feeding back to anatomical experiments.

Applying the method to the multi-scale network model of macaque visual cortex presented in Chap. 3, we derive targeted modifications of a set of critical connections, bringing the model closer to experimental observations of cortical activity. Based on the global properties of the bistable phase space of the model, the method refines the model's construction principles within experimental uncertainties and identifies the connections that decisively shape the dynamics. Preliminary results have been presented in abstract form (Schmidt et al., 2015b).

4.2 Materials and Methods

Consider a neuronal network of leaky integrate-and-fire model neurons with exponentially decaying postsynaptic currents. In the diffusion approximation, the dynamics of the membrane potential V and synaptic current I_s is (Fourcaud & Brunel, 2002)

$$\begin{aligned} \tau_\mathrm{m}\frac{\mathrm{d}V}{\mathrm{d}t} &= -V + I_\mathrm{s}(t) \\ \tau_\mathrm{s}\frac{\mathrm{d}I_\mathrm{s}}{\mathrm{d}t} &= -I_\mathrm{s} + \mu + \sigma\sqrt{\tau_\mathrm{m}}\xi(t), \end{aligned} \tag{4.1}$$

where τ_m is the membrane time constant and τ_s the synaptic time constant, respectively. The membrane resistance has been absorbed into the definition of the current. The input spike trains are approximated by a current which fluctuates with variance σ^2 around the mean value μ where the noise fluctuations are drawn from a Gaussian process $\xi(t)$ satisfying $\langle\xi(t)\rangle = 0$ and $\langle\xi(t)\xi(t')\rangle = \delta(t-t')$. Whenever the membrane potential V crosses the threshold θ, the neuron emits a spike and V is set to the reset potential V_r, where it is clamped during τ_r. In the networks we consider, the neurons are structured into N populations interconnected with a connectivity matrix $\mathbf{W} = \mathbf{K} \circledast \mathbf{J}$, where K_{ij} is the number of incoming connections to a neuron in population i from population j, and J_{ij} is the efficacy of a single synapse in the respective connection. Here, $\circledast$ denotes element-wise multiplication, also called the Hadamard product (see Cichocki et al., 2009, for a consistent set of symbols for operations on matrices). Additionally, each neuron is driven by K_ext Poisson sources with rate ν_ext and synaptic efficacy J_ext. All neurons in one population have identical parameters, so that we can describe the network activity in terms of population-averaged firing rates ν_i due to population-dependent input μ_i, σ_i determined by the connectivity. Using the Fokker-Planck formalism, the stationary firing rates for each population i are given by (Fourcaud & Brunel, 2002)

$$\begin{aligned} \frac{1}{\nu_i} &= \tau_\mathrm{r} + \tau_\mathrm{m}\sqrt{\pi}\int_{\frac{V_r-\mu_i(\boldsymbol{A})}{\sigma_i(\boldsymbol{A})}+\gamma\sqrt{\frac{\tau_\mathrm{s}}{\tau_\mathrm{m}}}}^{\frac{\theta-\mu_i(\boldsymbol{A})}{\sigma_i(\boldsymbol{A})}+\gamma\sqrt{\frac{\tau_\mathrm{s}}{\tau_\mathrm{m}}}} e^{x^2}\,(1+\mathrm{erf}(x))\,\mathrm{d}x \\ &=: 1/\Phi_i(\boldsymbol{\nu}, \boldsymbol{A}) \end{aligned} \tag{4.2}$$

$$\mu_i(\boldsymbol{A}) = \tau_\mathrm{m}\sum_j K_{ij}J_{ij}\nu_j + \tau_\mathrm{m}K_\mathrm{ext}J_\mathrm{ext}\nu_\mathrm{ext} \tag{4.3}$$

$$\sigma_i^2(\boldsymbol{A}) = \tau_\mathrm{m}\sum_j K_{ij}J_{ij}^2\nu_j + \tau_\mathrm{m}K_\mathrm{ext}J_\mathrm{ext}^2\nu_\mathrm{ext}, \tag{4.4}$$

which is correct up to linear order in $\sqrt{\tau_\mathrm{s}/\tau_\mathrm{m}}$ and where $\gamma = |\zeta(1/2)|/\sqrt{2}$, with ζ denoting the Riemann zeta function (Abramowitz & Stegun, 1974). Here, $\boldsymbol{A}$ is chosen from the set of model

parameters $\{\boldsymbol{K}, \boldsymbol{J}, \nu_{\text{ext}}, \dots\}$. If $\boldsymbol{A}$ is a matrix, we vectorize it by concatenating its rows and indicate this by lower case, i.e., $\boldsymbol{a} = (a_{00}, a_{01}, \dots, a_{0N}, a_{10}, \dots, a_{1N}, a_{N0} \dots, a_{NN}) = \text{vec}(\boldsymbol{A}^{\text{T}})^{\text{T}}$ following Magnus & Neudecker (1995). If the chosen parameter is a scalar we denote it with a. We find the fixed points of (4.2) by solving the first-order differential equation (Wong & Wang, 2006)

$$\dot{\boldsymbol{\nu}} \coloneqq \frac{\mathrm{d}\boldsymbol{\nu}}{\mathrm{d}s} = \boldsymbol{\Phi}(\boldsymbol{\nu}, \boldsymbol{A}) - \boldsymbol{\nu}, \tag{4.5}$$

using the classical fourth-order Runge-Kutta method (RK4) with step size $h = 0.01$ where s denotes a dimensionless pseudo-time. Note that Eq. 4.5 does not reflect the real time evolution of the population rates, but rather is a mathematical method to obtain the system's fixed points. In contrast to Wong & Wang (2006) we do not only search for stable fixed points, but also use (4.5) to obtain unstable attractors (cf. Sec. 4.3), an idea originating from the study of simple attractor networks by Amit & Brunel (1997b, esp. their Fig. 2 and Eq. 12).

The non-linearity in the input-output relationship $\boldsymbol{\Phi}$ in combination with feedback connections in general can cause multi-stability in the network. In particular excitatory-excitatory loops cause the system Eq. 4.5 to exhibit multi-stable behavior in the stationary firing rates. Thus we require a method to control influence of these loops on the phase space of the network. As an illustration we first study the mechanism using the simple network architecture depicted in Fig. 4.2A: a population of excitatory neurons is coupled to itself with indegree K and is driven by external Poisson sources with the same indegree $K_{\text{ext}} = K$ and rate ν_{ext}. All connections have identical synaptic weights J. An increase in the external drive shifts the input-output relationship $\boldsymbol{\Phi}(\boldsymbol{\nu}, a = \nu_{\text{ext}})$ of this one-dimensional system (Fig. 4.2C) to the left. The bifurcation diagram is shown in Fig. 4.2E: for low ν_{ext} there is only one fixed point with low activity (LA). When increasing ν_{ext}, a pair of fixed points of which one is stable and the other is unstable emerges via a saddle-node bifurcation, leading to a bistable system. The second stable fixed point exhibits high firing rates, denoted as the high-activity (HA) state. For even higher values of ν_{ext}, the LA state loses stability.

In the bistable situation, the initial condition of (4.5) determines which fixed point the system settles in. However, studying the behavior for a particular initial condition is of minor interest, since the actual spiking network is a stochastic system which fluctuates around the fixed points of the deterministic system defined by (4.5). Even if we knew that (4.5) would relax to the LA fixed point for one particular initial condition, this would not necessarily imply that this state is indefinitely stable. Global stability is determined by the size of the basin of attraction of the LA fixed point. Suppose we want to increase the activity at the LA fixed point while maintaining its global stability. Adjusting the internal connectivity K to preserve

the basin of attraction enables us to increase the external drive ν_{ext}.

The equilibria given by the zeros of the velocity $\dot{\nu}$ in the bistable case are shown in Fig. 4.2F. An increase of the drive on the one hand increases the firing rate of the LA fixed point (inset) but on the other hand reduces its basin of attraction, indicated by the colored bars in the top left corner. For illustrative purposes, we extend the problem to two dimensions by splitting the excitatory population into two subpopulations of equal size (Fig. 4.2B), mimicking a loop between excitatory populations in the model of macaque visual cortex (see Sec. 4.3). The corresponding (symmetric) two-dimensional phase space is shown in Fig. 4.2D. The basin of attraction for the LA fixed point, limited by the separatrix (Strogatz, 1994), is reduced for increased external drive. Since we have a bistable system, there must be at least one unstable fixed point on the separatrix at the intersection of the nullclines, i.e., the subspace for which the velocity $\dot{\nu}_i$ in direction i vanishes. We use the unstable fixed point to control the basin of attraction. To this end, we study the behavior of the fixed points with respect to an infinitesimal change $\boldsymbol{\delta a} = \boldsymbol{a'} - \boldsymbol{a}$ in the chosen model parameter. Let $\boldsymbol{\nu^*}(\boldsymbol{a})$ and $\boldsymbol{\nu^*}(\boldsymbol{a'})$ be the corresponding locations of the fixed points and $\boldsymbol{\delta\nu^*} = \boldsymbol{\nu^*}(\boldsymbol{a'}) - \boldsymbol{\nu^*}(\boldsymbol{a})$ their separation. We can then expand $\boldsymbol{\nu^*}(\boldsymbol{a'})$ into a Taylor series up to first order in $\boldsymbol{\delta a}$ and obtain

$$\begin{aligned} \boldsymbol{\nu^*}(\boldsymbol{a'}) &= \boldsymbol{\nu^*}(\boldsymbol{a}) + \boldsymbol{\Delta_a}\boldsymbol{\delta a} \\ \Leftrightarrow \boldsymbol{\delta\nu^*} &= \boldsymbol{\Delta_a}\boldsymbol{\delta a}\,, \end{aligned} \tag{4.6}$$

with

$$\begin{aligned} \Delta_{a,ij} &= \frac{\mathrm{d}\nu_i\left(\mu_i,\sigma_i\right)}{\mathrm{d}a_j} \\ &= \frac{\mathrm{d}\Phi_i\left(\mu_i,\sigma_i\right)}{\mathrm{d}a_j} \\ &= \underbrace{\frac{\partial\Phi_i}{\partial\mu_i}}_{S_i}\frac{\mathrm{d}\mu_i}{\mathrm{d}a_j} + \underbrace{\frac{\partial\Phi_i}{\partial\sigma_i}\frac{1}{2\sigma_i}}_{T_i}\frac{\mathrm{d}\sigma_i^2}{\mathrm{d}a_j}, \end{aligned} \tag{4.7}$$

where we notice that S_i and T_i only depend on the target population i. We accordingly define two diagonal matrices $\boldsymbol{S}$ and $\boldsymbol{T}$ with $S_{ii} = S_i$ and $T_{ii} = T_i$. The derivatives with respect to a_j have the compact expressions

$$\begin{aligned} \frac{\mathrm{d}\mu_i}{\mathrm{d}a_j} &= \frac{\partial\mu_i}{\partial a_j} + \sum_n \frac{\partial\mu_i}{\partial\nu_n}\frac{\mathrm{d}\nu_n}{\mathrm{d}a_j} \\ &= \left(\mathbf{D_a}\boldsymbol{\mu}\right)_{ij} + \tau_{\mathrm{m}}\sum_n \underbrace{\left(K_{in}J_{in}\right)}_{=W_{in}}\Delta_{a,nj}\,, \end{aligned}$$

with the Jacobian $(\mathbf{D}_{\boldsymbol{a}}\mathbf{f})_{ij} \coloneqq \frac{\partial f_i}{\partial a_j}$ of some vector-valued function $\mathbf{f}$ and

$$\frac{\mathrm{d}\sigma_{\mathrm{i}}^2}{\mathrm{d}a_j} = \left(\mathbf{D}_{\boldsymbol{a}}\boldsymbol{\sigma}^2\right)_{ij} + \tau_{\mathrm{m}} \sum_n (\underbrace{K_{in}J_{in}^2}_{=W_{2,in}})\Delta_{a,nj}\,,$$

where we use the Hadamard product again to define the matrix $\mathbf{W_2} \coloneqq \mathbf{K} \circledast \mathbf{J} \circledast \mathbf{J}$. Inserting the total derivatives into Eq. 4.7, we derive the final expression for $\boldsymbol{\Delta}_{\boldsymbol{a}}$, reading

$$\begin{aligned} \boldsymbol{\Delta}_{\boldsymbol{a}} &= \boldsymbol{S}\left[\mathbf{D}_{\boldsymbol{a}}\boldsymbol{\mu} + \tau_{\mathrm{m}}\boldsymbol{W}\boldsymbol{\Delta}_{\boldsymbol{a}}\right] + \boldsymbol{T}\left[\mathbf{D}_{\boldsymbol{a}}\boldsymbol{\sigma}^2 + \tau_{\mathrm{m}}\boldsymbol{W_2}\boldsymbol{\Delta}_{\boldsymbol{a}}\right] \\ \Leftrightarrow \boldsymbol{\Delta}_{\boldsymbol{a}} &= \left[\mathbb{1} - \underbrace{\tau_{\mathrm{m}}(\boldsymbol{S}\boldsymbol{W} + \boldsymbol{T}\boldsymbol{W_2})}_{=:M}\right]^{-1} \underbrace{\left[\boldsymbol{S}\mathbf{D}_{\boldsymbol{a}}\boldsymbol{\mu} + \boldsymbol{T}\mathbf{D}_{\boldsymbol{a}}\boldsymbol{\sigma}^2\right]}_{=:\bar{\boldsymbol{\Delta}}_{\boldsymbol{a}}}, \end{aligned} \tag{4.8}$$

where we use $\mathbb{1}$ for the identity matrix and define the effective connectivity matrix $\boldsymbol{M}$ and $\bar{\boldsymbol{\Delta}}_{\boldsymbol{a}}$. The latter is an $N \times P$ dimensional matrix, where P is the dimension of $\boldsymbol{a}$ (for example, $P = N^2$ for $\boldsymbol{a} = \boldsymbol{k}$ the vector of indegrees). With the aid of Eq. 4.8, evaluating Eq. 4.6 at the unstable fixed point predicts the shift of the separatrix (Fig. 4.2D) to linear order. We now consider an additional parameter $\boldsymbol{b}$ which is modified to counteract the shift of the unstable fixed point caused by the change in parameter $\boldsymbol{a}$, i.e.,

$$\begin{aligned} \boldsymbol{\Delta}_{\boldsymbol{a}}\boldsymbol{\delta a} &\stackrel{!}{=} -\boldsymbol{\Delta}_{\boldsymbol{b}}\boldsymbol{\delta b} \\ \Leftrightarrow \bar{\boldsymbol{\Delta}}_{\boldsymbol{a}}\boldsymbol{\delta a} &= -\bar{\boldsymbol{\Delta}}_{\boldsymbol{b}}\boldsymbol{\delta b}, \end{aligned} \tag{4.9}$$

where we used that the inverse of $\mathbb{1} - \boldsymbol{M}$ appears on both sides of the equation and hence drops out. Note that $(\boldsymbol{a}, \boldsymbol{b})$ represent any combination of model parameters, for example external input, indegrees, synaptic weights, etc. For a particular choice of $(\boldsymbol{a}, \boldsymbol{b})$ we solve Eq. 4.9 for $\boldsymbol{\delta b}$. For the illustrative example shown in Fig. 4.2, where $a = \nu_{\mathrm{ext}}$ and $b = K$, Eq. 4.9 simplifies to

$$\delta K = \frac{\bar{\Delta}_{\nu_{\mathrm{ext}}}\delta\nu_{\mathrm{ext}}}{\bar{\Delta}_K} = \frac{K_{\mathrm{ext}}\delta\nu_{\mathrm{ext}}}{\nu},$$

since the derivatives S and T appearing in the respective $\bar{\Delta}'$s cancel each other. The resulting velocity of the system $\Phi(a', b')$ (Φ defines the system Eq. 4.5) is shown in Fig. 4.2F. The firing rate in the LA fixed point is still increased as desired (inset), and the unstable fixed point coincides with that obtained in the original system $\Phi(a, b)$. This pattern of fixed points is also indicated by the zero vectors of the velocity field $\dot{\boldsymbol{\nu}}$ (Fig. 4.2D). The separatrix follows the unstable fixed point, and the basin of attraction in the system $\boldsymbol{\Phi}(a', b')$ is restored to that in the original system $\boldsymbol{\Phi}(a, b)$. Fig. 4.2G shows the behavior of the LA fixed point in more detail. The modification of the parameter b does not noticeably change the location of the LA fixed

point. In conclusion, the method allows us to increase the firing rates in the LA fixed point without modifying its basin of attraction.

To determine critical connections in a more complex model, we choose $\boldsymbol{b} = \boldsymbol{k}$, i.e. the vector of indegrees, and solve Eq. 4.9 with a decomposition into eigenmodes. We can write the right-hand side as

$$-\left(\bar{\boldsymbol{\Delta}}_{\boldsymbol{k}}\boldsymbol{\delta k}\right)_i = -\sum_j \tau_{\mathrm{m}}\left(S_i J_{ij} + T_i J_{ij}^2\right)\nu_j \delta K_{ij}\,. \tag{4.10}$$

The equation holds because μ_i, σ_i are only affected by connections to population i, and therefore their derivatives $\partial\mu_i/\partial k_l$, $\partial\sigma_i/\partial k_l$ and hence $\bar{\Delta}_{k,il}$, vanish for $l \notin [(i-1)N+1, iN]$. We now make the ansatz

$$\delta K_{ij} = \sum_l \frac{\epsilon_l}{\tau_{\mathrm{m}}} \frac{\left(\boldsymbol{u}^l \boldsymbol{v}^{l\mathrm{T}}\right)_{ij}}{\left(\boldsymbol{S}\boldsymbol{J} + \boldsymbol{T}(\boldsymbol{J} \circledast \boldsymbol{J})\right)_{ij}}\,, \tag{4.11}$$

which decomposes the changes $\boldsymbol{\delta K}$ into eigenmodes of the effective connectivity. The $\boldsymbol{u}^l$ and $\boldsymbol{v}^l$ are the l-th right and left eigenvectors of $\boldsymbol{M}$ as defined in Eq. 4.8, fulfilling the bi-orthogonality $\boldsymbol{v}^{l\mathrm{T}}\boldsymbol{u}^n = \delta_{ln}$ and the completeness relation $\sum_l \boldsymbol{u}^l\boldsymbol{v}^{l\mathrm{T}} = \mathbb{1}$. Inserting Eq. 4.10 with Eq. 4.11 into (4.9) yields

$$\bar{\boldsymbol{\Delta}}_{\boldsymbol{a}}\boldsymbol{\delta a} = -\sum_l \epsilon_l \boldsymbol{u}^l \boldsymbol{v}^{l\mathrm{T}}\boldsymbol{\nu}. \tag{4.12}$$

Thus we can solve for ϵ_l by multiplying from the left with $\boldsymbol{v}^{n\mathrm{T}}$

$$\begin{aligned}
\underbrace{\boldsymbol{v}^{n\mathrm{T}}\bar{\boldsymbol{\Delta}}_{\boldsymbol{a}}\boldsymbol{\delta a}}_{=:\hat{a}^n} &= -\sum_l \epsilon_l \underbrace{\boldsymbol{v}^{n\mathrm{T}}\boldsymbol{u}^l}_{\delta_{nl}}\underbrace{\boldsymbol{v}^{l\mathrm{T}}\boldsymbol{\nu}}_{=:\hat{\nu}^l} \\
\epsilon_n &= -\frac{\hat{a}^n}{\hat{\nu}^n}\,.
\end{aligned}$$

Our goal is to find a set of connections which dominate the size of the basin of attraction of the LA fixed point. Inserting Eq. 4.12 into Eq. 4.6 leads to

$$\begin{aligned}
\boldsymbol{\delta\nu}^* &= \sum_l (\mathbb{1} - \boldsymbol{M})^{-1}\epsilon_l \boldsymbol{u}^l \boldsymbol{v}^{l\mathrm{T}}\boldsymbol{\nu} \\
&= \sum_l \frac{\epsilon_l}{1-\lambda_l}\boldsymbol{u}^l\hat{\nu}^l \\
&= \sum_l \frac{-\hat{a}^l}{1-\lambda_l}\boldsymbol{u}^l,
\end{aligned}$$

where λ_l are the eigenvalues of $\boldsymbol{M}$, which are either real or complex conjugate pairs since $\boldsymbol{M} \in \mathbb{R}^{N\times N}$. To determine the influence of each eigenmode on the shift of the fixed point, we project the eigenvectors $\boldsymbol{u}^l$ onto the fixed-point shift $\boldsymbol{\delta\nu}^*$ by multiplying each side with

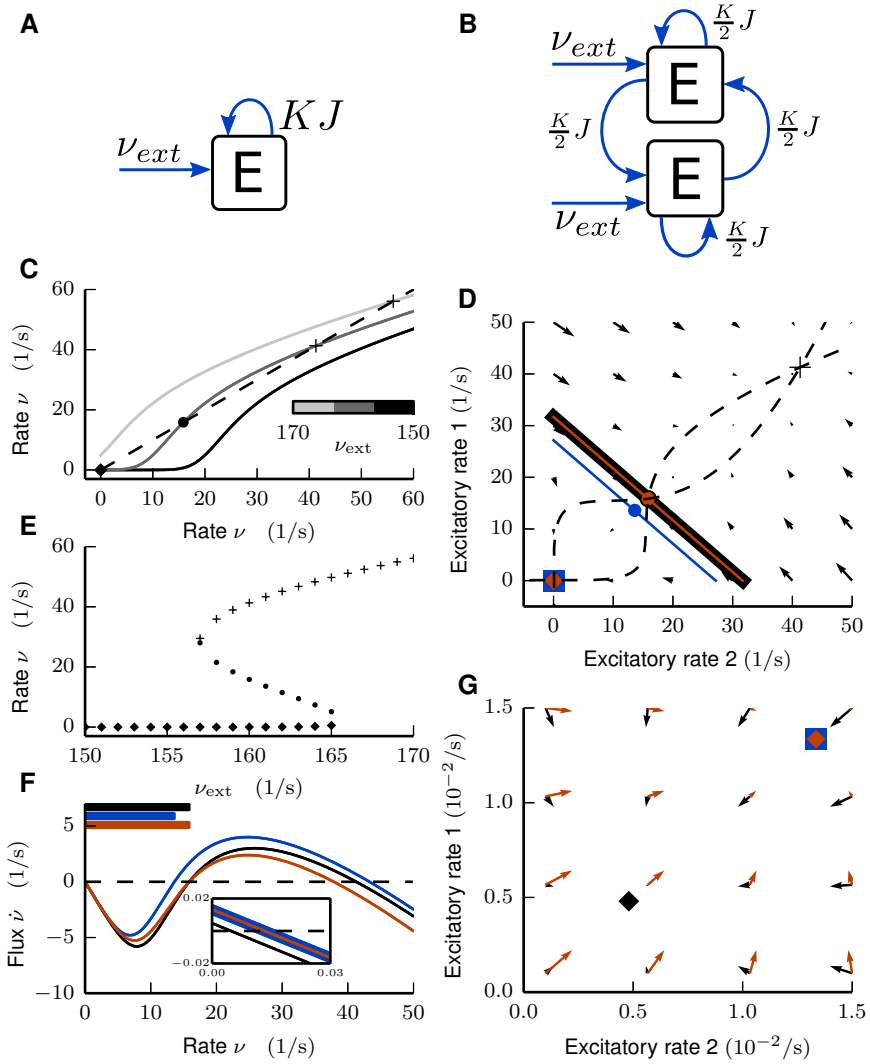

Figure 4.2: Activity flow in an illustrative network example. Left column: Global stability analysis in the single-population network. **A** Illustration of network architecture. **C** Input-output relationship $\Phi(\nu, \nu_{\text{ext}})$ for different rates of the external Poisson drive $\nu_{\text{ext}} = [150, 160, 170]\frac{1}{\text{s}}$ from black to light gray. Intersections with the identity line (dashed) mark fixed points of the system, which are shown in **E** as a function of ν_{ext}. **F** Flux $\dot{\nu}$ in the bistable case for $\Phi(\nu_{\text{ext}} = 160\frac{1}{\text{s}}, K)$ in black, $\Phi(\nu'_{\text{ext}} = 161\frac{1}{\text{s}}, K)$ in blue, and modified system $\Phi(\nu'_{\text{ext}} = 161\frac{1}{\text{s}}, K')$ in red. Intersections with zero (dashed line) mark fixed points. The inset shows an enlargement close to the LA fixed point. Horizontal bars at top of figure denote the size of the basin of attraction for each of the three settings. **Right column:** Global stability analysis in the network of two mutually coupled excitatory populations. **B** Illustration of network architecture. **D** Flow field and nullclines (dashed curves) for $\boldsymbol{\Phi}(\nu_{\text{ext}} = 160\frac{1}{\text{s}}, K)$ and separatrices (solid lines), LA fixed point (rectangle), HA fixed point (cross) and unstable fixed points (circles) for $\boldsymbol{\Phi}(\nu_{\text{ext}} = 160\frac{1}{\text{s}}, K)$ in black, $\boldsymbol{\Phi}(\nu_{\text{ext}} = 161\frac{1}{\text{s}}, K)$ in blue, and $\boldsymbol{\Phi}(\nu_{\text{ext}} = 161\frac{1}{\text{s}}, K')$ in red. The red separatrix and the red unstable fixed point coincide with the black ones. **G** Enlargement of D close to the LA fixed points. Flow field of original system shown in black, of modified system in red.

$\boldsymbol{\delta\nu}^*\boldsymbol{\delta\nu}^{*\mathrm{T}}$ and solve again for $\boldsymbol{\delta\nu}^*$ to obtain

$$\boldsymbol{\delta\nu}^* = \sum_l \underbrace{\frac{-\hat{a}^l}{1-\lambda_l} \frac{\boldsymbol{\delta\nu}^{*\mathrm{T}}\boldsymbol{u}^l}{\boldsymbol{\delta\nu}^{*\mathrm{T}}\boldsymbol{\delta\nu}^*}}_{=:\tilde{\eta}_l} \boldsymbol{\delta\nu}^*, \tag{4.13}$$

where we define the (possibly complex-valued) coefficients $\tilde{\eta}_l$. We aim at a decomposition of $\boldsymbol{\delta\nu}^*$ into real components. If $\mathrm{Im}(\lambda_l) = 0$, $\tilde{\eta}_l$ is real, so we can work directly with $\eta_l \coloneqq \tilde{\eta}_l$. Complex eigenvalues $\mathrm{Im}(\lambda_l) \neq 0$ and corresponding eigenvectors come in conjugate pairs, so in this case we combine the corresponding coefficients $\eta_l \coloneqq \tilde{\eta}_l + \tilde{\eta}_l^*$, to have all contributions $\eta_l \in \mathbb{R}$ and $\sum_l \eta_l = 1$ by construction Eq. 4.13. Each η_l quantifies how much of the total fixed-point shift can be attributed to the l-th eigenmode, which allows identification of the most effective eigendirection in the next section, where we apply the ansatz (4.11) to a multi-area, multi-layer model of macaque visual cortex.

The spiking simulations of the network model are carried out with the NEST simulator (Gewaltig & Diesmann, 2007) on the JUQUEEN supercomputer (Jülich Supercomputing Centre, 2015). All simulations can be reproduced with NEST 2.8.0, while the results presented in this study were performed with further optimizations for the supercomputer which will enter one of the next releases. All simulations use a time step of 0.1 ms and exact integration for the sub-threshold dynamics of the leaky integrate-and-fire neuron model (reviewed in Plesser & Diesmann, 2009).

4.3 Results

We investigate a multi-scale network of cortical areas to understand the structural features essential for a realistic state of baseline activity. The model extends and adapts the microcircuit model of Potjans & Diesmann (2014), which covers 1 mm^2 surface area of early sensory cortex (Fig. 4.3A), to all vision-related areas of macaque cortex (Fig. 4.3B). Based on the microcircuit model, an area is composed of 4 layers (2/3, 4, 5, and 6) each having an excitatory (E) and an inhibitory (I) neural population, except parahippocampal area TH which consists of only 3 layers (2/3, 5, and 6). From anatomical studies, it is known that cortical areas in the macaque monkey are heterogeneous in their laminar structure and can be roughly categorized into 8 different architectural types based on cell densities and laminar thicknesses. This distinction was originally developed for prefrontal areas (Barbas & Rempel-Clower, 1997), and then extended to the entire cortex (Hilgetag et al., 2016). The visual cortex, and thus the model, comprises areas of categories 2,4,5,6,7 and 8. Precise layer-specific neuron densities are available for a number of areas, while for other areas, the neuron density is estimated based on their architectural type

(see Schmidt et al. (2015a) for details).

The inter-areal connectivity is based on binary data from the CoCoMac database (Stephan et al., 2001; Bakker et al., 2012; Felleman & Van Essen, 1991; Rockland & Pandya, 1979; Barnes & Pandya, 1992) indicating the existence of connections, and quantitative data from Markov et al. (2014a). The latter are retrograde tracing data where connection strengths are quantified by the fraction of labeled neurons (FLN) in each source area. The parcellation into different areas of our model follows the FV91 scheme proposed by Felleman & Van Essen (1991). The original analysis of the experimental data was performed in a different parcellation scheme, the M132 atlas of Markov et al. (2014a). Both parcellations have been registered to the F99 space (Van Essen, 2002), a standard macaque cortical surface included with the software tool Caret (Van Essen et al., 2001). This enables mapping between the two parcellations M132 and FV91.

On the target side, we use the exact coordinates of the injections to identify the equivalent area in the FV91 parcellation. To map the data on the source side from the M132 atlas to the FV91 parcellation, we count the number of overlapping triangles on the F99 surface between any given pair of regions and distribute data proportionally to the amount of overlap using the F99 region overlap tool on http://cocomac.g-node.org. In the model, this FLN is mapped to the indegree K_{AB} the target area A receives from source area B divided by its total indegree, i.e., $FLN_{AB} = K_{AB} / \sum_{B'} K_{AB'}$. Here, K_{AB} is defined as the total number of synapses between A and B divided by the total number of neurons in A. On the source side, laminar connection patterns are based on CoCoMac (Felleman & Van Essen, 1991; Barnes & Pandya, 1992; Suzuki & Amaral, 1994b; Morel & Bullier, 1990; Perkel et al., 1986; Seltzer & Pandya, 1994) and on fractions of labeled neurons in the supragranular layers (SLN, Markov et al., 2014b). Gaps in these data are bridged exploiting a sigmoidal relation between SLN and the logarithmized ratio of overall cell densities of the two areas. We map the SLN to the ratio between the number of synapses originating in layer 2/3 and the total number of synapses between the two areas, assuming that only excitatory populations send inter-area connections, i.e., $SLN_{AB} = \sum_i K_{AB}^{i,2/3\mathrm{E}} N_A^i / \sum_{i,j} K_{AB}^{ij} N_A^i$, where the indices i and j go over the different populations within area A and B, respectively. On the target side, the CoCoMac database provides data from anterograde tracing studies (Felleman & Van Essen, 1991; Rockland & Pandya, 1979; Jones et al., 1978; Seltzer & Pandya, 1991). A detailed description of the data integration is given in Schmidt et al. (2015a).

Simulations of the model (Fig. 4.3C) reveal that, though realistic levels of activity can be achieved for populations in layers 2/3 and 4, populations 5E and 6E of the majority of areas show vanishingly low and zero activity in contrast to empirical data (Swadlow, 1988; de Kock & Sakmann, 2009). Missing inputs in the model, i.e., from subcortical and non-visual cortical areas, are replaced by Poissonian spike trains, whose rate ν_{ext} is a free, global parameter. In

the original model all populations of a particular area receive the same indegree of external inputs K_{ext}. The only exception to this rule is area TH where the absence of granular layer 4 is compensated by an increase of the external input to populations 23E and 5E by 20 %. To elevate the firing rates in the excitatory populations in layers 5 and 6, we enhance the external Poisson drive, parametrized by the $K_{5E,\text{ext}}$ incoming connections per target neuron (indegree), of the 5E population by increasing $\kappa = K_{\text{5E,ext}}/K_{\text{ext}}$. The simultaneous increase in the drive of 6E needs to be stronger, since the firing rates in population 6E of the original model (Fig. 4.3C) are even lower than the rates of 5E (averaged across areas: 0.1 spikes/s for 5E compared to $2 \cdot 10^{-4}$ spikes/s for 6E). We thus scale up $K_{\text{6E,ext}}$ linearly with κ such that $\kappa = 1.15$ results in $K_{\text{6E,ext}}/K_{\text{ext}} = 1.5$. However, already a perturbation of a few percent leads to a state with unrealistically high rates (Fig. 4.3D), caused by the reduced basin of attraction of the low activity state similar to Fig. 4.2D. Our aim now is to improve the working point of the model such that all populations exhibit spiking activity $\gtrsim 0.05$ spikes/s while preventing the model from entering a state of unrealistically high rates of $\gtrsim 30$ spikes/s (Swadlow, 1988; de Kock & Sakmann, 2009). The employed technique exposes the mechanism giving rise to the observed instability and identifies the circuitry responsible for this dynamical feature.

In the following we choose the inactive state $\boldsymbol{\nu}(0) = (0, \ldots, 0)^{\text{T}}$ as the initial condition. The exact choice is not essential since we are only interested in the fixed points of the system. Fig. 4.4B shows the integration of Eq. 4.5 over pseudo-time s for different increased levels of the external drive to populations 5E and 6E parametrized by κ. For low values of κ, the integration converges to the LA fixed point shown in Fig. 4.4D, and is in agreement with the activity emerging in the simulation (Fig. 4.3C). For increased values of κ, the system settles in the HA fixed point (Fig. 4.4E), again in agreement with the simulation. The population-specific firing rates in the HA state found in the mean-field predictions (Fig. 4.4E) are also close to those in the simulation (Fig. 4.3D), but minor deviations occur due to the violation of the assumptions made in the diffusion approximation. In particular, at these pathologically high rates, the neurons fire regularly and close to the reciprocal of their refractory period, while in the mean-field theory we assume Poisson spike statistics. Still, the mean-field theory quantitatively predicts the bistability found in the simulation. Since the theory yields reliable predictions in both stable fixed points, we assume that also the location of the unstable fixed point in between these two extremes is accurately predicted by the theory.

To stabilize the LA fixed point for increased values of the external drive κ, we apply the procedure derived in Sec. 4.2. To this end we need to find the unstable fixed point of the system. This is nontrivial since the high-dimensional system of equations Eq. 4.2 is not directly solvable for the fixed-point firing rates, and the numerical integration of Eq. 4.5 for finding equilibria by construction only converges to stable fixed points. If the unstable fixed

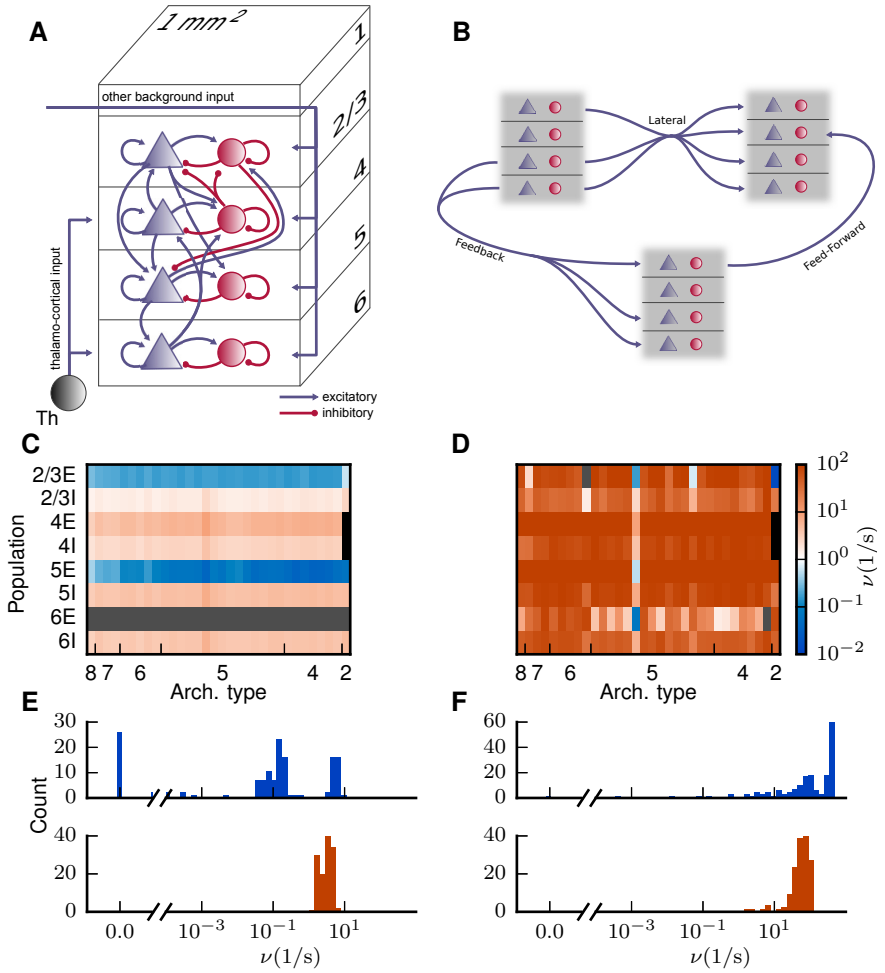

Figure 4.3: Bistability of the model. **A** Sketch of the microcircuit model serving as a prototype for the areas of the multi-area model (adapted from Potjans & Diesmann, 2014, with permission). **B** Sketch of the most common laminar patterns of cortico-cortical connectivity of the multi-area model. **C** Population-averaged firing rates encoded in color for a spiking network simulation of the multi-area model with low external drive ($\kappa = 1.0$). **D** As **C** but for increased external drive ($\kappa = 1.125$). The color bar refers to both panels. Areas are ordered according to their architectural type along the horizontal axis from V1 (type 8) to TH (type 2) and populations are stacked along the vertical axis. The two missing populations 4E and 4I of area TH are marked in black and firing rates $< 10^{-2}$ Hz in gray. **E** Histogram of population-averaged firing rates shown in **C** for excitatory (red) and inhibitory (blue) populations. The horizontal axis is split into linear- (left) and log-scaled (right) ranges. **F** as **E** corresponding to state shown in **D**.

point has only one repelling direction (Fig. 4.6A), it constitutes a stable attractor on the $N-1$ dimensional separatrix. The separatrix is a stable manifold (Strogatz, 1994), and therefore a trajectory originating in its vicinity but not near an unstable fixed point initially stays in the neighborhood. If an initial condition just outside the separatrix is close to the basin of attraction of a particular unstable fixed point, the trajectory initially approaches the latter. Close to the fixed point the velocity is small. Ultimately trajectories diverge from the separatrix in the fixed point's unstable direction, as illustrated in Fig. 4.4A. In conclusion, we expect a local minimum in the velocity along the trajectories close to the unstable fixed point. To estimate the location of the unstable fixed point in this manner, we need to find initial conditions close to the separatrix. Naively, we would just fix the value of κ and vary the initial condition. However, due to the high dimensionality of our system, this is not feasible in practice, and we vary κ for a fixed initial condition instead (Fig. 4.4B), which is expected to yield approximately the same result. Fig. 4.4B shows the firing rate averaged across populations for two trajectories starting close to the separatrix, whereby the first one converges to the LA fixed point and the second one to the HA state. The trajectories diverge near the unstable fixed point and thus we define the last local minimum of the Euclidean norm of the velocity vector as the critical time s_c at which we assume the system to be close to the unstable fixed point (Fig. 4.4C). We find four relevant and distinct unstable fixed points, of which two are shown in Fig. 4.5.

To counteract the shift of the separatrix caused by the increase of κ, we follow the procedure described in Sec. 4.2. We subject the modifications of connectivity following from the application of the method to the following constraints. In line with the anatomical literature, we do not allow for changes of the connectivity that would lead to cortico-cortical connections originating in the granular layer 4 (Felleman & Van Essen, 1991), and we also disallow inhibitory cortico-cortical connections, as the vast majority of long-range connections are known to be excitatory (Salin & Bullier, 1995; Tomioka & Rockland, 2007). In addition, we naturally restrict indegrees to positive values. We find that four iterations (numbered by index j) corresponding to the four distinct unstable fixed points suffice to preserve the basin of attraction of the LA state with respect to an increase of the external drive up to $\kappa = 1.15$. In the following we concentrate on the iterations 1 and 2, where the second one is also representative for iterations 3 and 4, which are qualitatively alike. To derive the required modifications of the indegree matrix, we decompose $\boldsymbol{K}$ into its N eigenmodes and quantify the contribution of each eigenmode to the shift of the unstable fixed point (see Sec. 4.2). This allows us to identify the most effective eigendirection: in each iteration j there is exactly one unstable eigendirection with an eigenvalue $\mathrm{Re}\left(\lambda_\mathrm{c}^{(j)}\right) > 1$ (Fig. 4.6A). The associated critical eigenvector is approximately anti-parallel to the shift of the fixed point, $\boldsymbol{\delta\nu}^*$ (inset of Fig. 4.6B), and of similar length. The critical eigendirection (red dot in Fig. 4.6B) constitutes the most effective modification, giving

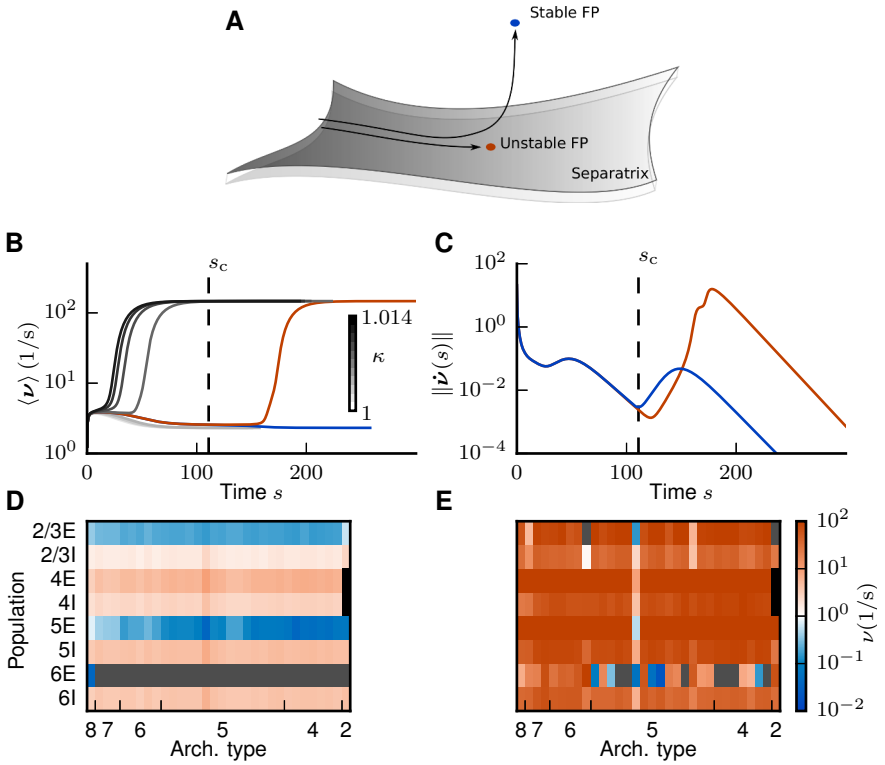

Figure 4.4: Application of the mean-field theory to the multi-area model of visual cortex. A Trajectories starting inside the separatrix converge to an unstable fixed point. Trajectories starting close to the separatrix are initially attracted by the unstable fixed point but then repelled in the fixed point's unstable direction and finally converge to a stable fixed point. **B** Firing rate averaged across populations over time. Integration of Eq. 4.5 leads to convergence to either the low-activity (LA) or the high-activity (HA) attractor for different choices of the external input factor κ. We show eight curves with κ varying from 1.0 to 1.014 in steps of 0.002 and two additional curves for $\kappa = 1.007662217, 1.007662218$. The curves for the largest factor ($\kappa = 1.007662217$) that still leads to the LA state and for the smallest factor ($\kappa = 1.007662218$) that leads to the HA state are marked in blue and red, respectively. The four curves with $\kappa \leq 1.006$ coincide with the blue curve. **C** Euclidean norm of the velocity vector in the integration of Eq. 4.5 for the different choices of κ. The vertical dashed line indicates the time s_c of the last local minimum in the blue curve. **D** Stationary firing rate in the different areas and layers of the model in a low-activity state for $\kappa = 1.0$ as predicted by the mean-field theory (same display as in Fig. 4.3). **E** As **D**, but showing the high-activity state for $\kappa = 1.125$.

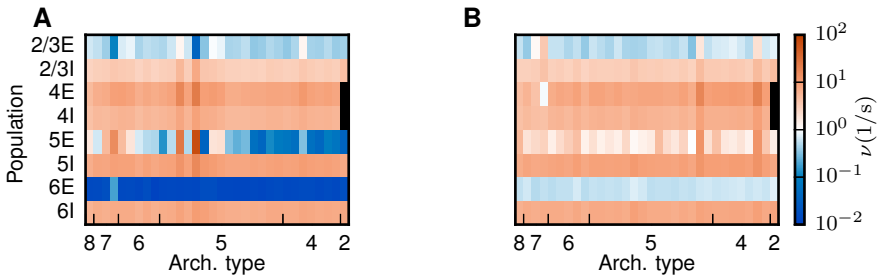

Figure 4.5: Unstable fixed points in subsequent iterations. Population firing rates at the unstable fixed point as predicted by the mean-field theory encoded in color for iterations 1 (**A**) and 2 (**B**). Same display as in Fig. 4.3.

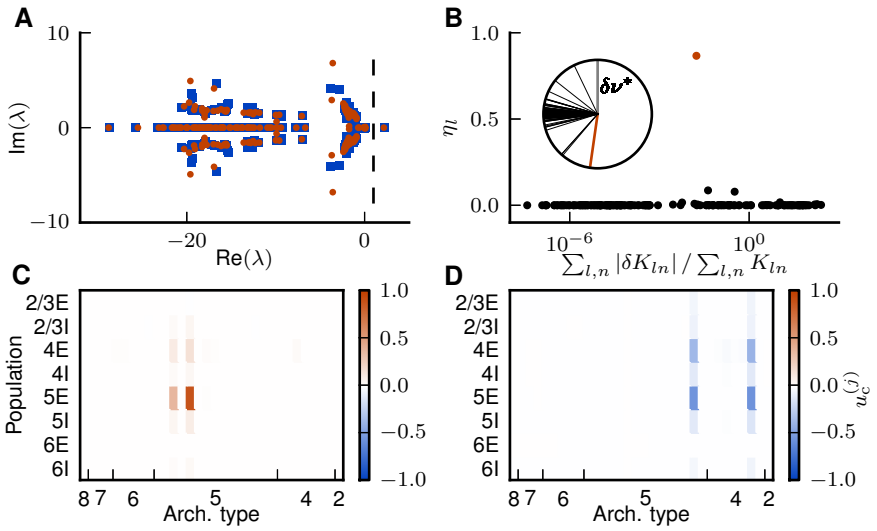

Figure 4.6: Eigenspectrum analysis of network stability. **A** Eigenvalue spectrum of the effective connectivity matrix M for the first (blue squares) and second (red dots) iteration. The dashed vertical line marks the edge of stability at a real part of 1. **B** Contribution η_l of an individual eigenprojection l to the shift of the unstable fixed point Eq. 4.13 versus the relative change in indegrees associated with l for the first iteration. The data point corresponding to $\lambda_c^{(1)}$ is marked in red. The inset shows the relative angles between $\boldsymbol{\delta\nu}^*$ and the eigenvectors $\boldsymbol{u}^l$. The red line corresponds to the critical eigendirection. **C** Entries of the eigenvector $u_c^{(1)}$ associated with $\lambda_c^{(1)}$ in the populations of the model. The affected areas are 46 and FEF. **D** Same as **C** for the second iteration.

the largest contribution to the desired shift while requiring only a small change of 2.3 % in average total indegrees.

The associated eigenvector $u_c^{(1)}$ predominately points into the direction of populations 4E and 5E of areas FEF and 46 (Fig. 4.6C), while $u_c^{(2)}$ has large entries in the 5E populations of two areas (Fig. 4.6D). The high rates of these populations at the unstable fixed points (cf. Fig. 4.5A,B with Fig. 4.6C,D) reflect that the instability is caused by increased rates in excitatory populations. Considering the other iterations (not shown) as well, these instabilities occur particularly in population 5E. Each iteration shifts the transition to the HA state (the value of κ for which the separatrix crosses the initial condition) to higher values of κ and increases the feasible rates of populations 5E and 6E in the LA state (Fig. 4.7A). After all four iterations, the average total indegrees (summed over source populations) of the system are changed by 11.3%. The first iteration mainly affects connections within and between areas 46 and FEF (Fig. 4.7B). In particular, the excitatory loops between the two areas are reduced in strength, especially those involving layer 5 (Fig. 4.7C). We thus identify two areas forming a critical loop. In the remaining iterations, the changes are spread across areas and especially connections originating in layer 5 are weakened (Fig. 4.7D). In conclusion, the method identifies critical structures in the model both on the level of areas and on the level of layers and populations, and leads to a small but specific structural change of the model.

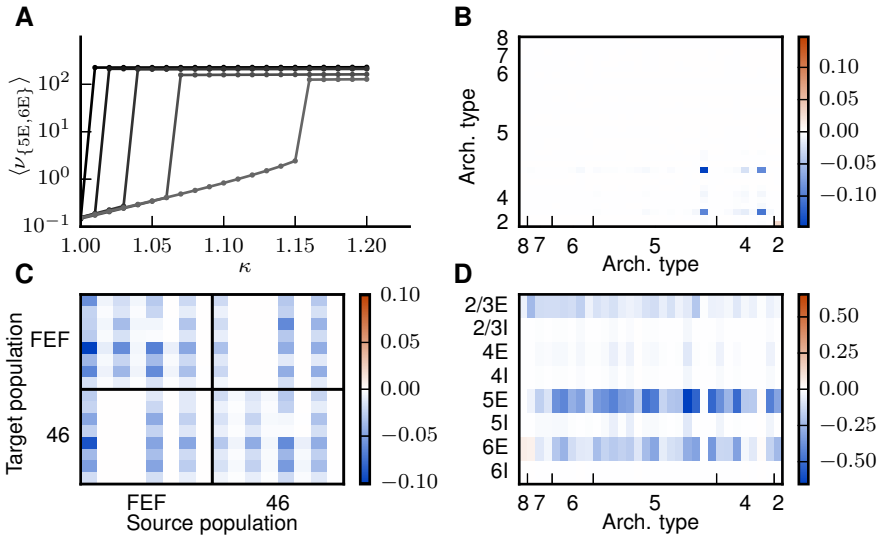

Figure 4.7: Altered phase space and modified connections. **A** Firing rates averaged across populations 5E and 6E and across areas for different stages from the original model (black) to iteration 4 (light gray) as a function of κ, predicted by the mean-field theory. **B** Relative changes in the indegree $\delta K_{AB} / \sum_B K_{AB}$ between areas A, B in the first iteration. **C** Layer-specific relative changes $\delta K_{ln} / \sum_n K_{ln}$ in the connections within and between areas FEF and 46, for the first iteration. Populations are ordered from 2/3E (left) to 6I (right) on the horizontal axis and from 6I (bottom) to 2/3E (top) on the vertical axis as in panel **D**. **D** Relative changes in population-specific indegrees summed over target populations, $\sum_l \delta K_{ln} / \sum_l K_{ln}$, combined for iterations two, three and four.

In the following we analyze the modifications of the connectivity with respect to the internal and inter-area connections in detail. The intrinsic circuits of the areas are modified in different directions, as shown for two exemplary areas V4 and CITv in Fig. 4.8A. Despite this heterogeneity, significant changes affect mostly excitatory-excitatory connections (Fig. 4.8A) with connections to population 5E experiencing the most significant changes (Fig. 4.7B,D). In fact, the anatomical data (Binzegger et al., 2004) underlying the microcircuit model (Potjans & Diesmann, 2014) contain only two reconstructed excitatory cells from layer 5, but considerably more for other cell types, indicating a higher uncertainty for layer 5 connections. Fig. 4.8B shows the correlation between intrinsic connectivity changes for all pairs of areas, with areas ordered according to a hierarchical clustering using a farthest point algorithm (Voorhees, 1986) on the correlation matrix. We find 4 clusters each indicating a group of areas which undergo changes with similar patterns. The groups are displayed in different colors in the histogram in Fig. 4.8B. Areas with architectural type 4, 5 and 6 are distributed over several clusters. We can interpret this as a differentiation of these types into further subtypes. The resulting changes of the intra-areal connectivity are small (Fig. 4.8A), but still significant for network stability.

Connections between areas can be characterized by their FLN and SLN. The FLN reflects the overall strength of an inter-areal connection and is only weakly affected across connections (Fig. 4.8C), with a correlation between original and modified logarithms of FLN

of $r_{\text{Pearson}} = 0.79$. Significant variations in the FLN occur mostly for very weak connections where the relative uncertainties in the experimental data are likely to be substantial. The two overlapping red dots in Fig. 4.8C represent the connections between areas 46 and FEF, which are modified in the first iteration (cf. Fig. 4.7C). The SLN determines the laminar pattern of the location of source neurons for cortico-cortical connections. Overall, data are available for 24% of the inter-areal connections in the parcellation of Felleman & Van Essen (1991), while the SLN for the rest are derived from the sigmoidal law. The majority of connections undergo small changes in their laminar source pattern (Fig. 4.8D) and connections with large modifications ($|\delta SLN| > 0.5$) are weak (average $\overline{FLN} = 6 \cdot 10^{-4}$ compared to $\overline{FLN} = 10^{-2}$ in the model as a whole). Because weak connections are represented by low neuron counts, they have a relatively large uncertainty in their laminar patterns, justifying larger adjustments. Spearman's rank correlation between the SLN of the original model that were directly taken from experiments and the logarithmized ratios of cell densities is $\rho = -0.63$ ($p = 3 \cdot 10^{-11}$, p-value of a two-sided test for uncorrelated data). For the modified model, we take the SLN of all connections into account and obtain $\rho = -0.40$ ($p = 6 \cdot 10^{-20}$), indicating a reduced, but still significant, monotonic dependence between SLN and the logarithmized ratios of cell densities.

We quantify the variability of the cortico-cortical connections (Markov et al., 2014a) as the average inter-individual standard deviation of the logarithmic FLN, i.e., $\sigma = \left\langle \sqrt{\overline{\left(\log FLN - \overline{\log FLN}\right)^2}} \right\rangle$, where the overbar $\bar{\cdot}$ denotes the average over injections and $\langle \cdot \rangle$ the average over connections. This variability equals 2.17 while the average modification of the logarithmic FLN is 1.34. The main experimental connection probabilities used to construct the intra-areal connectivity of the model have an average relative standard deviation of 30% across electrophysiological experiments (cf. Table 1 of Potjans & Diesmann, 2014) while the intra-areal connection probabilities of the model are modified by 9% on average. Scannell et al. (2000) report even greater variability in a review on cortico-cortical and thalamocortical connectivity. These considerations show that the changes applied to the connectivity are well within the uncertainties of the data. Overall, 35 out of 603 connections were removed from the network. In the CoCoMac database, 83 % of these are indicated by only a single tracer injection, while the overall proportion of connections measured by a single injection is 59 %.

Simulation of the full spiking network model with the modified connectivity matrix and $\kappa = 1.125$ to avoid being too close to the transition (4.7A). This results in reasonable rates across populations and areas (Fig. 4.9B, D). The average rates in populations 5E and 6E are increased compared to a simulation of the original model from 0.09 and $2 \cdot 10^{-5}$ spikes/s to 3.0 and 0.4 spikes/s, respectively. All populations exhibit firing rates within a reasonable range between 0.05 and 30 spikes/s (Fig. 4.9D), as opposed to the original state in which a considerable fraction of excitatory neurons was silent (Fig. 4.3E). The theoretical prediction is generally in

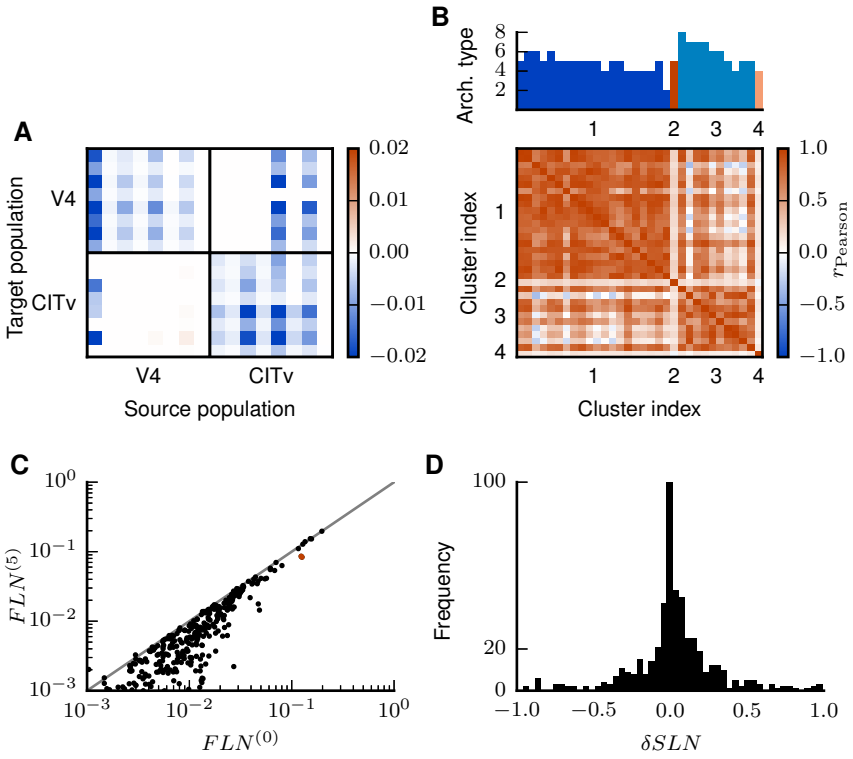

Figure 4.8: Analysis of changes in connectivity. **A** Changes in the indegrees within and between exemplary areas V4 and CITv relative to the total indegrees of the target populations, i.e., $\delta K_{ln} / \sum_n K_{ln}$. Populations ordered as in Fig. 4.7C. **B** Pearson correlation coefficient of the changes of the internal indegrees δK_{AA}^{ij} between all pairs of the 32 areas. Areas ordered according to hierarchical clustering using a farthest point algorithm (Voorhees, 1986). The heights of the bars on top of the matrix indicate the architectural types of the areas (types 1 and 3 do not appear in the model) with color representing the respective clusters. **C** FLN of the modified connectivity after 4 iterations versus the original FLN of the model. Only $FLN > 10^{-3}$ are shown for a better overview. The overlapping red dots represent the connections between areas 46 and FEF. Unity line shown in gray. **D** Histogram of the changes in SLN ($\delta SLN = SLN^{(4)} - SLN^{(0)}$).

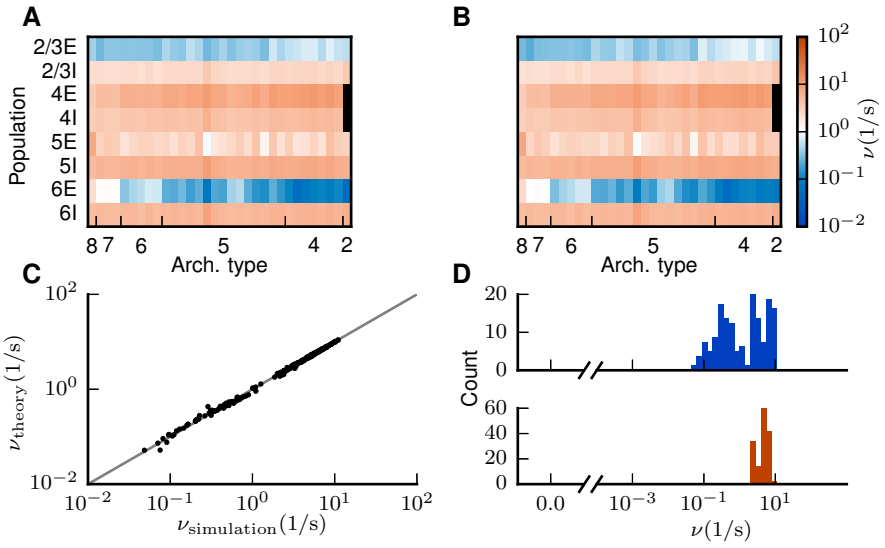

Figure 4.9: Improved low-activity fixed point of the model. Population-averaged firing rates for $\kappa = 1.125$ encoded in color (**A**) predicted by the analytical theory and (**B**) obtained from the full simulation of the spiking network. Same display as in Fig. 4.3. **C** Analytical versus simulated firing rates (black dots) and identity line (gray). **D** Histogram of population-averaged simulated firing rates. Same display as in Fig. 4.3.

agreement with the rates obtained in the simulation (Fig. 4.9A, C). Discrepancies are caused by violations of the employed assumptions, i.e., Poissonian spiking statistics (Fourcaud & Brunel, 2002). Moreover, we observe broad distributions of firing rates over neurons in the simulated data, seemingly in conflict with the reduction to population-averaged firing rates used by our framework. Nevertheless, the differences between theory and simulation are small and negligible for the central aim of the study: the integration of activity constraints into the data-driven construction of multi-scale neuronal networks.

4.4 Conclusion

In this chapter, we have investigated the link between experimentally measured structural connectivity and neuronal activity in the multi-scale spiking network model of macaque visual cortex constructed in Chap. 3. We devise a theoretical method that systematically combines anatomical connectivity with physiological activity constraints. Already the weak constraint of the activity neither vanishing nor being pathologically high yields a set of specific but minimal structural modifications for increasing the model's excitation to a realistic level. The procedure constrains the experimentally obtained connectivity maps to a realization that is compatible with physiological experiments. This establishes a path from the experimentally observed activity to specific hypotheses about the anatomy and demands for further experiments. In the next chapter, we analyse the dynamical behavior of the model using the modified connectivity

matrix.

CHAPTER 5

Dynamical results of the spiking network model

The following chapter has been published as preprint in Schmidt M, Bakker R, Shen K, Bezgin G, Hilgetag C-C, Diesmann M, and van Albada SJ (2015). Full-density multi-scale account of structure and dynamics of macaque visual cortex. arXiv:1511.09364 (Schmidt et al., 2015a). Preliminary results have been presented in Schmidt et al. (2013) and Schmidt et al. (2014).

Author contributions: Under supervision of Sacha J. van Albada, Rembrandt Bakker and Markus Diesmann, the author performed all simulations, analyzed and interpreted the results in collaboration with the co-authors.

5.1 Introduction

In this chapter, we analyze the properties of the connectivity graph derived in Chap. 3 and Chap. 4 and investigate the dynamical behavior of the model. To this end, we perform numerical simulations (see Sec. 6.5.1) and analyze them with the methods detailed in Sec. 6.5.2.

5.2 Community structure of anatomy relates to functional organization

An analysis of community structure in the weighted and directed graph of area-level connectivity reveals 6 clusters (Fig. 5.1). We test the significance of the corresponding modularity $Q = 0.32$ by comparing with 1000 surrogate networks conserving the total outdegree of each area by shuffling its targets. This yields $Q = -0.02 \pm 0.03$, indicating the significance of our clustering. The community structure reflects anatomical and functional properties of the areas. Two large clusters comprise ventral and dorsal stream areas, respectively. Ventral area VOT is grouped with early visual area VP. Early sensory areas V1 and V2 form a separate cluster, as well as parahippocampal areas TH and TF. The two frontal areas FEF and 46 form the last cluster. Nonetheless, the clusters are heavily interconnected (Fig. 5.1).

5.3 Area- and population-specific activity in the resting state

The model with cortico-cortical synaptic weights equal to local weights displays a reasonable ground state of activity but no substantial inter-area interactions (Fig. 6.3). To control these interactions, we scale cortico-cortical synaptic weights w_{cc} onto excitatory neurons by a factor $\lambda = J^{\mathcal{E}}_{\mathrm{cc}}/J$ and provide balance by increasing the weights $J^{\mathcal{I}}_{\mathrm{cc}}$ onto inhibitory neurons by twice this factor, $J^{\mathcal{I}}_{\mathrm{cc}} = \lambda_{\mathcal{I}}\lambda J = 2\lambda J$. In the following, we choose $\lambda = 1.9$. Simulations yield irregular activity with plausible firing rates (Fig. 5.2A-C). Irregularly occurring population bursts arise from the asynchronous baseline activity (Fig. 5.2G) and propagate across the network. The firing rates differ across areas and layers and are generally low in L2/3 and L6 and higher in L4 and L5, partly due to the cortico-cortical interactions (Fig. 5.2D). The overall average rate is 17.2 spikes/s. Inhibitory populations are generally more active than excitatory ones across layers and areas despite the identical intrinsic properties of the two cell types. However, the strong participation of L5E neurons in the cortico-cortical interaction bursts causes these to fire more rapidly than L5I neurons. Pairwise correlations are low throughout the network (Fig. 5.2E).

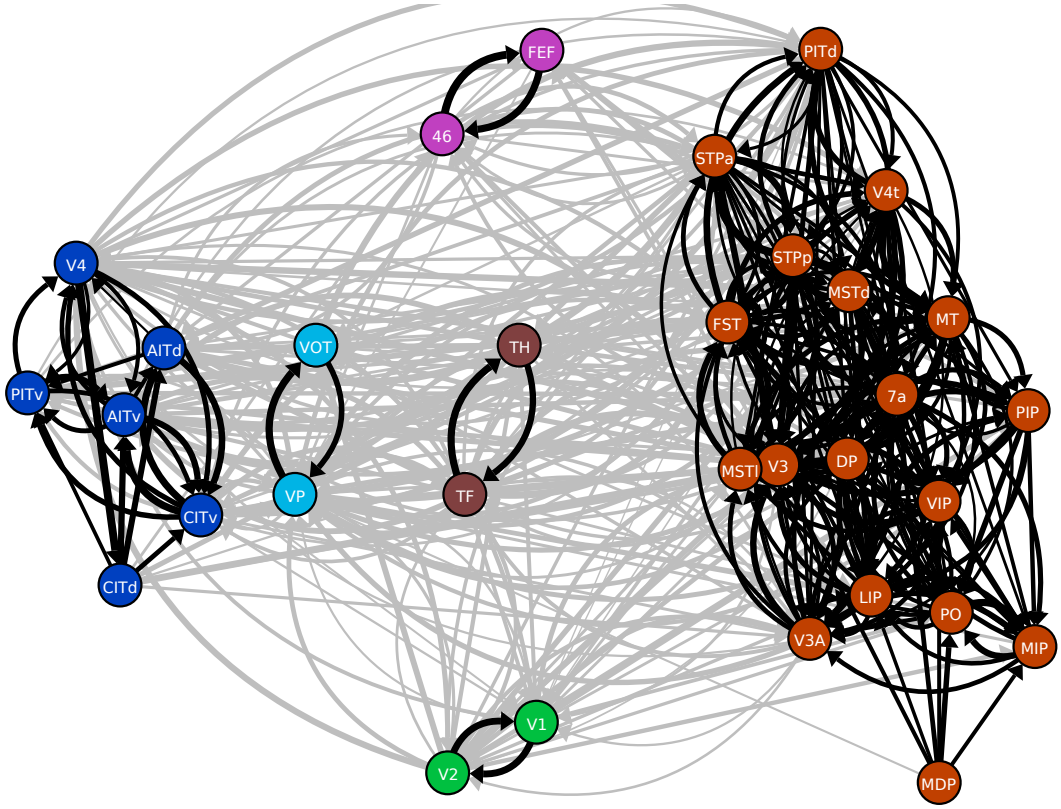

Figure 5.1: Community structure of the model. Clusters in the connectivity graph detected by the map equation method (Rosvall et al., 2010), indicated by the color of the nodes: Early visual areas (green), dorsal stream areas (red), areas VP and VOT (light blue), ventral stream (dark blue), parahippocampal areas (brown), and frontal areas (purple). Black, connections within clusters; gray, connections between clusters. Line thickness encodes logarithmized outdegrees. Only edges with relative outdegree$> 10^{-3}$ are shown.

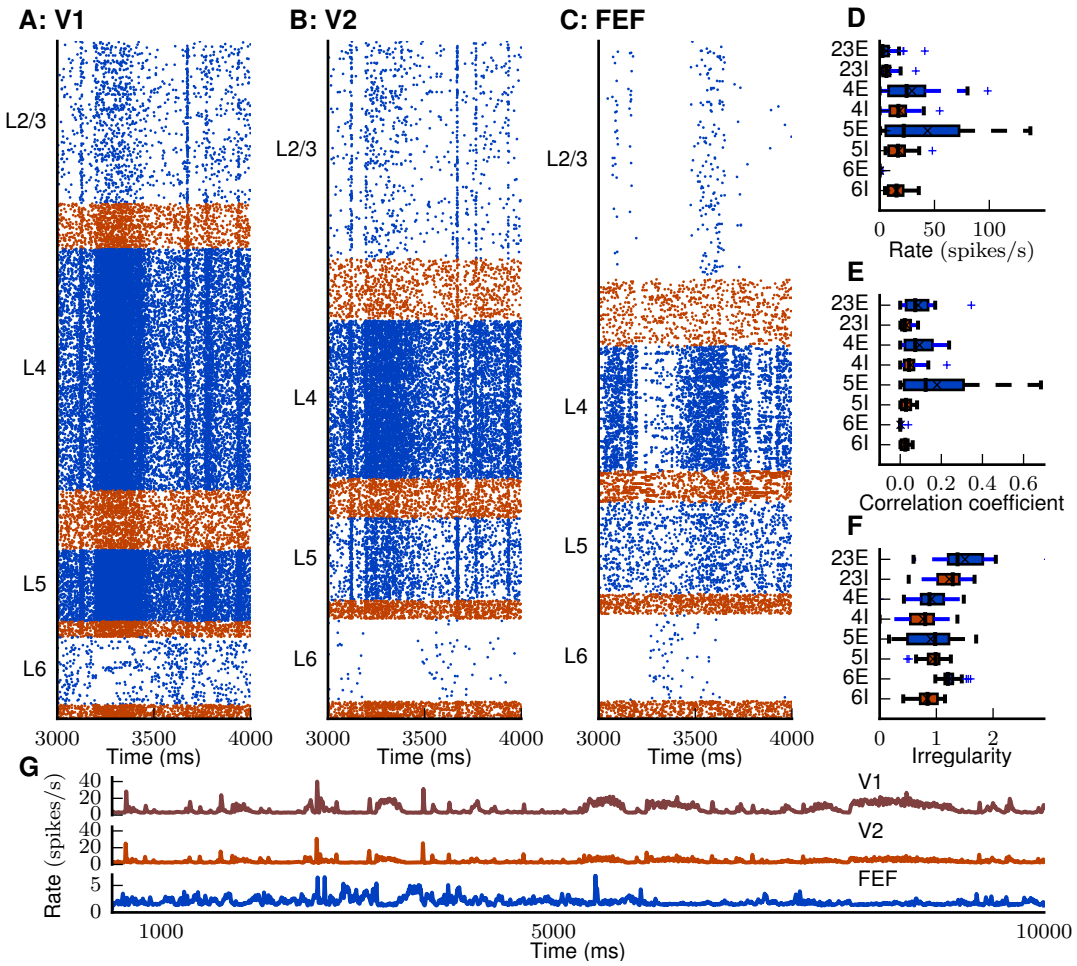

Figure 5.2: Resting state of the model. (A-C) Raster plot of spiking activity of 3 % of the neurons in area V1 (A), V2 (B), and FEF (C). (D-F) Spiking statistics across areas and populations shown as area-averaged box plots. (D) Population-averaged firing rates. (E) Average pairwise correlation coefficients of spiking activity. (F) Irregularity measured by revised local variation LvR (Shinomoto et al., 2009) averaged across neurons. (G) Area-averaged firing rates.

Excitatory neurons are more synchronized than inhibitory cells in the same layer, except for L6. Spiking irregularity is close to that of a Poisson process across areas and populations, with excitatory neurons consistently firing more irregularly than inhibitory cells (Fig. 5.2F). Higher areas exhibit bursty spiking, as illustrated by the raster plot for area FEF (Fig. 5.2C).

5.4 Intrinsic time scales increase with anatomical hierarchy

We tested whether the model accounts for the hierarchical trend in intrinsic time scales observed in macaque cortex (Murray et al., 2014). Indeed, autocorrelation width in the model

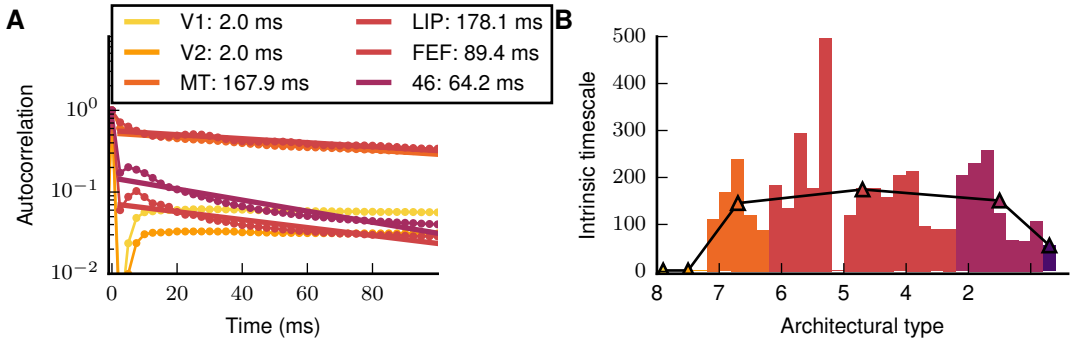

Figure 5.3: Intrinsic time scales. (A) Dots, autocorrelation function averaged across all populations in each area (colors in legend). Lines, exponential fit $f(t) = A \cdot \exp[-t/\tau]$. (B) Intrinsic time scales of areas vs. architectural type (color-coded). Average time scale per architectural type indicated by triangles and overall trend by black curve. Area MDP (architectural type 5) has a time scale of 2 ms because it is uncoupled from other areas due to the lack of incoming connections.

increases from early visual to higher areas. In early visual areas including V1, the autocorrelation decays with $\tau < 2.5$ ms, indicating near-Poissonian spiking (Fig. 5.3A). In higher areas, autocorrelations are broader with decay times on the order of 100 ms. Note that the single-cell auto-correlations differ from the auto-correlations of the compound population signals, which are dominated by cross-correlations since the number of neuron pairs far exceeds the number of neurons (Kriener et al., 2008). In contrast to the single cells, the compound population signals do not show an increase in time scales along the structural hierarchy (Fig. 6.4).

The long time scales reflect bursty spike patterns of single-neuron activity (Fig. 5.2). A simulation with equal intrinsic and long-range synaptic weights that showed no significant interactions yields near-Poissonian spiking in all areas (Fig. 6.3), showing that the cortico-cortical interactions elicit the increased time scales. The fact that the areas differ mainly in their average indegrees suggests that the longer time scales in higher areas reflect their dense connectivity, caused by lower neuron densities and the constant synapse density. Area 46, which overlaps with lateral prefrontal cortex studied by Murray et al. (2014), shows a shorter time scale compared to the experimental data. However, in line with Murray et al. (2014), we find the time scale of area LIP to exceed that of MT, albeit by a small amount.

5.5 Structural and hierarchical directionality of spontaneous activity

To investigate inter-area propagation, we determine the temporal order of spiking (Fig. 5.4A) based on the correlation between areas. We detect the location of the extremum of the correlation function for each pair of areas (Fig. 5.4B) and collect the corresponding time lags in a matrix (Fig. 5.4C). In analogy to structural hierarchies based on pairwise connection patterns

(Reid et al., 2009), we look for a temporal hierarchy that best reflects the order of activations for all pairs of areas (see Sec. 6.5.2). The result (Fig. 5.4E) places parietal and temporal areas at the beginning and early visual as well as frontal areas at the end. The first and second halves of the time series yield qualitatively identical results (Fig. 6.5). Fig. 5.4D shows the consistency of the hierarchy with the pairwise lags. To quantify the goodness of the hierarchy, we counted the pairs of areas for which it indicates a wrong ordering. The number of such violations is 190 out of 496, well below the $230 \pm 12\,(\mathrm{SD})$ violations obtained for 100 surrogate matrices, created by shuffling the entries of the original matrix while preserving its antisymmetric character. This indicates that the simulated temporal hierarchy reflects nonrandom patterns. The propagation is mostly in the feedback direction not only in terms of the structural hierarchy, but also spatially: activity starts in parietal regions, and spreads to the temporal and occipital lobes (Fig. 5.5). However, activity troughs in frontal areas follow peaks in occipital activity and appear last.

5.6 Emerging interactions mimic experimental functional connectivity

We compute the area-level functional connectivity (FC) based on the synaptic input current to each area, which has been shown to be more comparable to the BOLD fMRI than the spiking output (Logothetis et al., 2001). The FC matrix exhibits a rich structure, similar to experimental resting-state fMRI (Fig. 5.6A, B, see Chap. 3 for details). In the simulation, frontal areas 46 and FEF are more weakly coupled with the rest of the network, but the anticorrelation with V1 is well captured by the model (Fig. 6.6). Area MDP sends connections to, but does not receive connections from other areas according to CoCoMac, limiting its functional coupling to the network. Louvain clustering (Blondel et al., 2008), an algorithm optimizing the modularity of the weighted, undirected FC graph (Newman, 2004), yields two modules for both the simulated and the experimental data. We denote the simulated modules by 1S, 2S and the experimental ones by 1E, 2E. The modules from the simulation combine different structural clusters and differ in terms of their average position in the temporal hierarchy shown in Fig. 5.4C. Cluster 1S merges early visual with ventral and two dorsal regions with average level in the temporal hierarchy of $\overline{h} = 0.47 \pm 0.13\,(\mathrm{SD})$. Cluster 2S contains mostly temporally earlier areas ($\overline{h} = 0.33 \pm 0.25\,(\mathrm{SD})$) merging parahippocampal with dorsal but also frontal areas. The experimental module 2E comprises only dorsal areas, while 1E consists of all other areas including also eight dorsal areas.

The structural connectivity of our model shows higher correlation with the experimental

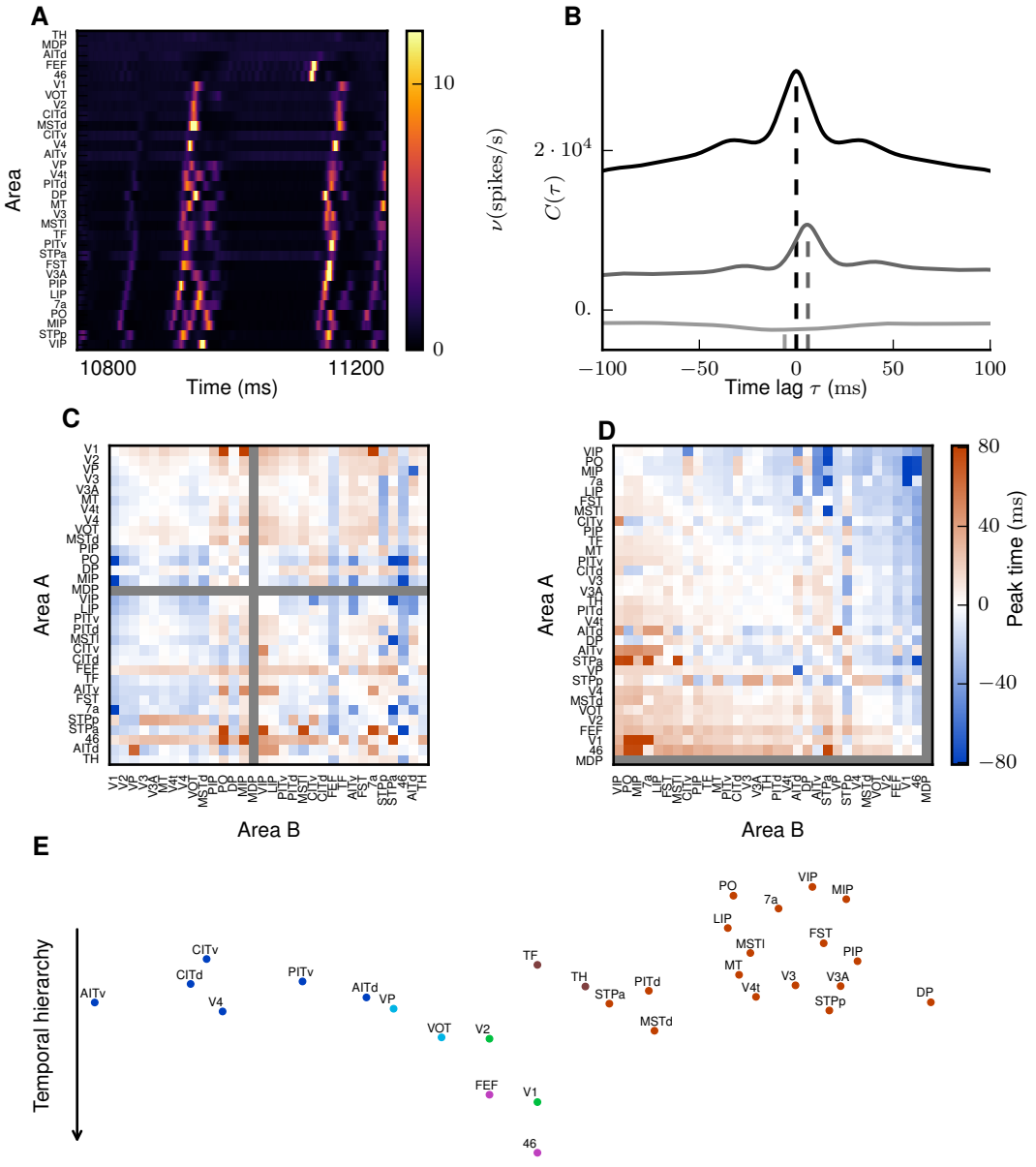

Figure 5.4: Temporal hierarchy. (A) Area-averaged firing rates for a sample period, with areas ordered according to the onset of increased activity from $t > 1250$ ms. (B) Covariance functions of the area-averaged firing rates of V1 with areas V2 (gray) and FEF (light gray), and auto-covariance function of V1 (black). Dashed lines mark selected time lags, detected by a wavelet smoothing algorithm (see Sec. 6.5.2). (C) Matrix of time lags of the correlation function for all pairs of areas. Area MDP was neglected because it has only outgoing connections to but no incoming connections from other visual areas according to CoCoMac. (D) Temporal hierarchy. Colors correspond to the map equation clustering (cf. Fig. 5.1). Areas are horizontally arranged to avoid visual overlap. (E) Peak position matrix with areas in hierarchical order.

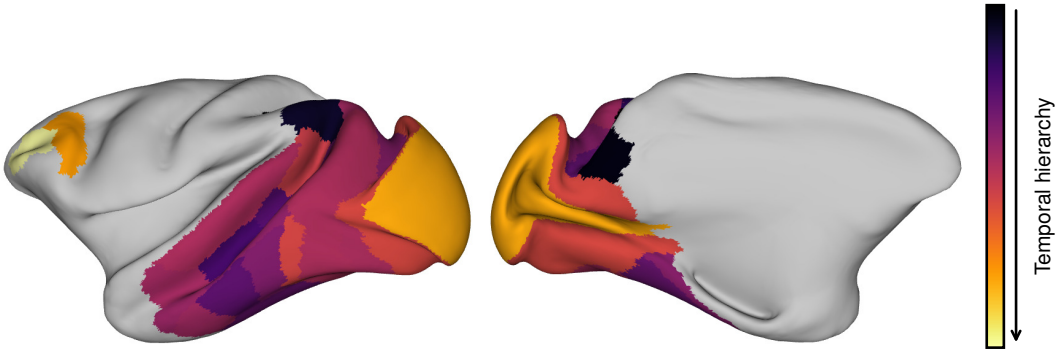

Figure 5.5: Temporal hierarchy on the cortical surface. Lateral (left) and medial (right) view on the left hemisphere of an inflated macaque cortical surface showing the order in which areas are preferentially activated. Created with the "view/map 3d surface" tool on http://scalablebrainatlas.incf.org.

FC ($r_{\text{Pearson}} = 0.34$) than the binary connectivity matrices from both a previous (Shen et al., 2015) and the most recent release of CoCoMac ($r_{\text{Pearson}} = 0.20$), further validating our weighted connectivity matrix. For increasing weight factor λ, the correlation between simulation and experiment improves (Fig. 5.6D). For $\lambda = 1$, areas interact weakly, resulting in low correlation between simulation and experiment (Fig. 6.3). For intermediate cortico-cortical connection strengths, the correlation of simulation vs. experiment exceeds that between the structural connectivity and experimental FC (Fig. 5.6C), indicating the enhanced explanatory power of the dynamical model. From $\lambda = 2$ on, the network is prone to switch to the high-activity state (Fig. 6.7). Thus, the highest correlation ($r_{\text{Pearson}} = 0.47$ for $\lambda = 1.9$) occurs just below the onset of a state in which the model visits both the low-activity and high-activity attractors.

5.7 Conclusion

In this chapter, we have analyzed the dynamical behavior of the network model that we constructed and refined in Chap. 3 and Chap. 4, respectively. We find the model to exhibit a stable ground state with area- and population-specific activity. Increasing the synaptic weights of cortico-cortical connections leads to structured interactions between areas that trigger long intrinsic time scales in higher areas, reminiscent of experimental findings (Murray et al., 2014). The inter-area interactions follow structured patterns close to experimental resting-state data exceeding the similarity of the structural connectivity with the experiment. Moreover, we observe activity to propagate predominantly in feedback direction with the exception of the frontal areas, which are anti-correlated with early visual areas, in line with experimental findings.

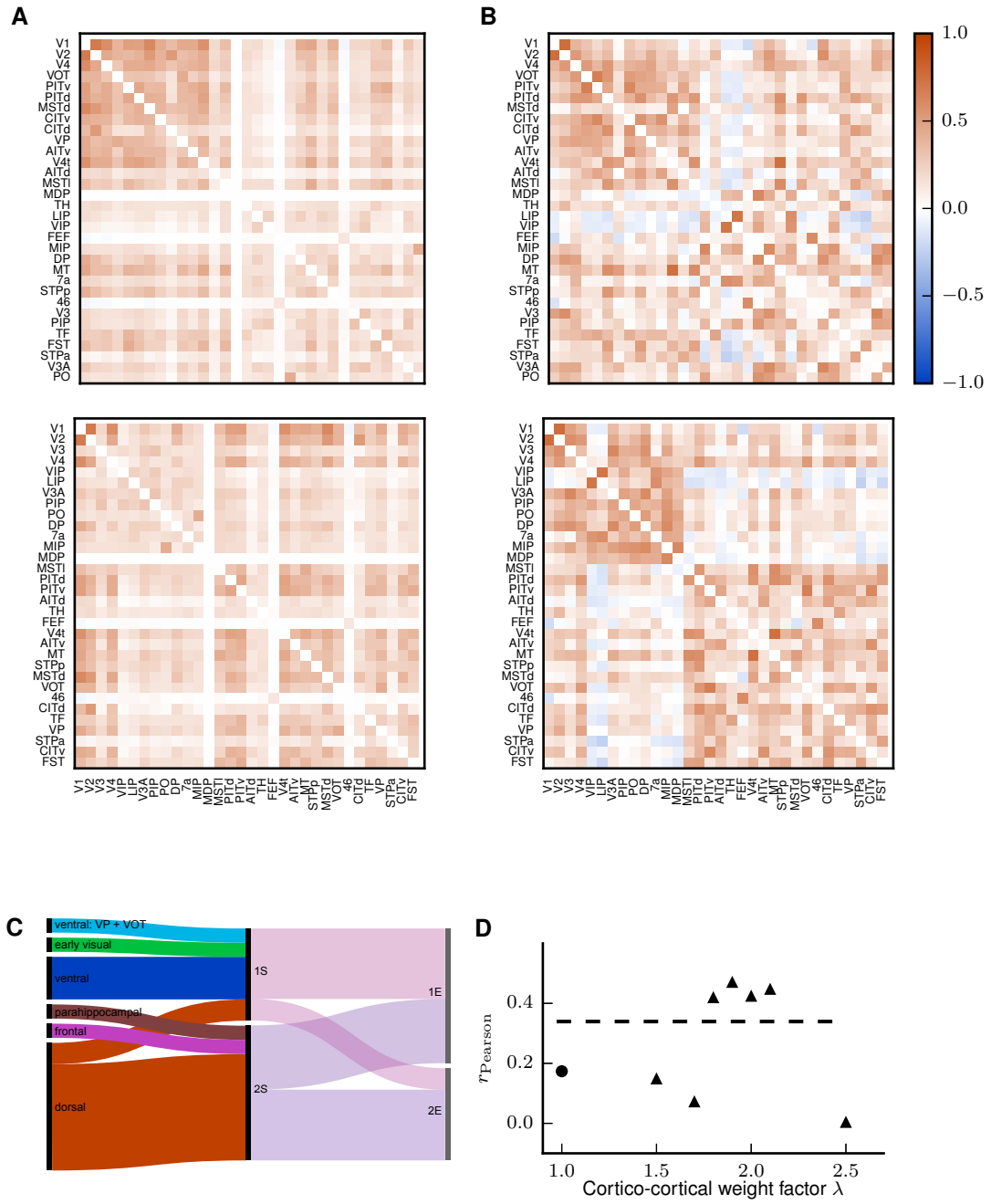

Figure 5.6: Inter-area interactions. (A) Simulated functional connectivity (FC) for $\lambda = 1.9$ measured by the zero-time lag correlation coefficient of synaptic input currents. (B) FC from macaque resting-state fMRI (see Chap. 3). Areas are ordered according to a clustering with the Louvain algorithm (Blondel et al., 2008) applied to the simulated data (top row) and to the experimental data (bottom row), respectively (see Chap. 3). (C) Alluvial diagram showing the differences in the clusters for the structural connectivity (left), the simulated FC (center) and the experimentally measured FC (right). (D) Pearson correlation coefficient of simulated FC vs. experimentally measured FC for varying λ with $\lambda_{\mathcal{I}} = 2$ (triangles) and $\lambda_{\mathcal{I}} = 1$ (dot, cf. Fig. 6.3). Dashed line, Pearson correlation coefficient of structural connectivity vs. experimentally measured FC.

III

Discussion

In this thesis, we present solutions to problems at different stages in the workflow of large-scale spiking network simulations. The fourth generation kernel of the network simulation software NEST lays the foundations of full-scale simulations of cortical networks exploiting the full compute power of modern supercomputers. In the present study, the reduced memory footprint of the connection infrastructure enabled us to simulate the network model on smaller machine sizes to save resources. We have compiled data from a large body of experimental studies on macaque cortex and constructed a multi-scale network model of macaque visual cortex in a systematic manner that allows for iterative refinement by incorporation of further details on cortical structure. Our theoretical method enables us to predict the attractors of the network model, analyze their relation to the network structure, and extend the model definition process to experimental data on cortical dynamics. The network model features a realistic ground state of ongoing cortical activity and reproduces features of cortical dynamics at multiple scales.

The new software kernel of the open-source software NEST enables computational neuroscientists to simulate cortical networks at full scale on modern computer systems to circumvent the problems of down-scaling such networks and study multi-scale cortical dynamics. The presented multi-area model provides a path from detailed anatomical data to a consistent definition of a multi-scale network model and demonstrates the feasibility of large-scale simulations on supercomputers by reproducing essential experimental findings of cortical dynamics. The theoretical method aids neuroscientists in defining and constraining such models. Furthermore, it suggests a new approach to gain insights into cortical networks by generating predictions on network structure. The following sections will discuss the results in detail.

Spiking network simulation code for petascale computers

In Chap. 2, we present the development of the fourth generation kernel of the NEST simulator. NEST is a general-purpose simulator for neural networks, scalable from laptops to supercomputers. In contrast to many other software tools that are run on supercomputers, its structures are not specifically optimized for the use on supercomputers, but are designed to enable scalability for machine sizes over several orders of magnitude. We achieve this by using object-oriented programming techniques such as recursive C++ templates and polymorphism to implement a flexible, adaptive infrastructure. Concretely, the connection framework is formulated as templates with the synapse type and target identifier as template arguments. This separates the simulator infrastructure from the implementation of the synapse dynamics, facilitating the extension of the simulator with new synapse types. The random network that is used for the benchmarks represents a worst-case scenario for a network simulator due to its

lack of local structure that could be exploited in the distribution of neurons over the machine nodes.

NEST distributes neurons over machine nodes in a round-robin fashion. In general, this balances the workload between the machine nodes, but passes up the chance to decrease communication between nodes by placing heavily interconnected clusters of neurons on the same machine. Moreover, Kunkel et al. (2012a) studied a network where a fraction of the incoming synapses has a local origin and showed that exploiting this by placing locally interconnected groups of neurons on the same machines leads to a decrease of memory consumption. The new adaptive infrastructure reduces this advantage by enabling efficient storage of short target lists.

The new infrastructure reduces the memory footprint of a network simulation by a factor of four and leads to better performance, most likely due to more efficient cache usage. This has two complementary advantages: on a given machine size, NEST can represent larger networks, thus enabling neuroscientists to study networks of more realistic sizes. On the other hand, a given neuronal network can be represented on smaller machines, leading to a reduction of the required resources, which are typically restricted on shared computer clusters. On the entire K computer, a world record was achieved by simulating $1.86 \cdot 10^9$ neurons connected via $11.1 \cdot 10^{12}$ synapses using the entire effectively available memory of 1.07 PB of RAM (RIKEN BSI, 2013). Building the network took 800 s and simulating 1 s of biological time required 2500 s. On JUQUEEN, the new technology led to a maximal network of $1.08 \cdot 10^9$ neurons with $6.5 \cdot 10^{12}$ synapses, which is approximately the size of a cat's brain. The performance improvements are not restricted to supercomputers: Simulating the benchmark model with $11,250$ neurons and $67.5 \cdot 10^6$ synapses on a laptop for 1 s took 35.5 s with both NEST 3g and the 4g kernel, but the memory consumption decreased from 5.25 GB with NEST 3g to 3.11 GB with NEST 4g.

The performance of software is often evaluated by measuring the number of executed floating-point operations (flops) per second. For a distributed network simulator like NEST, a large part of the workload is caused by random access of memory to deliver spike events. The simulator spends on the order of 10 flops per time step and neuron. In the extreme case, when a software uses fixed-point arithmetic or for a pure data base application, the number of flops per second would be zero. To reasonably judge the performance of a software, it would thus be necessary to normalize the measured number of flops per second to the actual amount of flops in the code. Another way to assess software performance is through strong scaling benchmarks, where a fixed problem size is evaluated for varying machine sizes. This measure does not reflect the typical use of a network simulator, where it is not the time for solving the differential equations that limits the problem size per core but the storage of the synapses. Moreover, strong scaling results can be misleading. For instance, improving the solvers for the neuronal state variables and thus decreasing the workload of single cores can lead to a

seemingly worse curve because the communication overhead will already be visible for smaller machine sizes. The same argument is true for weak scaling, where the machine size is varied while keeping the problem size per machine constant. Here, the optimal case is a constant runtime versus machine size. The scaling plot can be improved by simulating more neurons per core to increase the workload of a CPU, masking the communication overhead. Moreover, if the memory overhead grows with machine size, it is necessary to adapt the problem size to the largest machine size, thereby decreasing the workload on small machine sizes. In our study, we have used a maximum-filling benchmark to evaluate the performance of the software, i.e., we measured the largest possible problem size for varying machine sizes. For NEST 4g, this is almost equivalent to a weak scaling (Fig. 2.9). In the case of NEST 3g, the problem size scales sub-linearly with machine size (Fig. 2.7). The 3g kernel therefore has a worse weak scaling performance just because the workload per machine has to be decreased, making the communication overhead even more severe. The maximum filling benchmark thus allows for better comparability between the two kernels. Moreover, a NEST simulation usually requires a minimum of resources (in core hours) if the smallest possible machine size is used. The maximum filling benchmark therefore measures the minimal resources for a given problem. By normalizing the simulation time by the workload on each core, i.e., by the number of neurons per core, one may gain deeper insight into the performance of the simulator. The problems of benchmarking neural simulators have been discussed in more detail by van Albada et al. (2014).

The presented pool allocator constitutes a good solution for the large-scale scenario of a supercomputer. It does not, however, allow for the reallocation of memory. For the large-scale case, this is not of relevance because with high probability, any neuron has either zero or one local targets on each machine so that the simulator does not have to switch from case 1 to case 2 (cf. Fig. 2.3). For intermediate scenarios, this will frequently happen because neurons are likely to have several local targets, thus requiring reallocation of memory in the target lists. The pool allocator should thus be switched off, as it is done by default. The downside of this solution is that the user has to choose an allocator at compile time. For small and mid-scale scenarios, one possibility could be to precompute the number of local targets. This would, however, prevent involving mechanisms such as structural plasticity, where synapses are removed and created at runtime, and degeneration of synapses.

The presented technology can be generalized to other network simulators under three assumptions. First, a single neuron is entirely represented on a single machine. Second, synapses are stored on the side of the postsynaptic neurons. Third, spikes are communicated globally between the single cores. In applications with communication schemes that use directed communication, i.e., a core receives only spikes from neurons with local targets, it is possible to remove the sparse table. On the other hand, more infrastructure on the presynaptic side is re-

quired in this case. The C2 simulator (Ananthanarayanan & Modha, 2007) uses such a directed communication scheme. In NEURON (Hines et al., 2008) and the Neural Tissue Simulator (Kozloski & Wagner, 2011), neurons can be spread over multiple cores, which would require a major revision of the data structures. In general, the presented software solution could be useful for any computational problem that involves the interaction of a fixed number of heterogeneous components on numbers of cores varying on multiple orders of magnitude.

Construction of the spiking network model

In Chap. 3, we present the definition of the spiking network model, resulting in an area- and population-specific connectivity map. We construct the cortico-cortical connectivity by combining binary connectivity information from the CoCoMac database (Stephan et al., 2001; Bakker et al., 2012) with quantitative data from retrograde tracing studies (Markov et al., 2014a,b). Using tracing data is beneficial in comparison to information gained by diffusion tensor imaging (DTI) because it provides more detailed data, e.g., laminar information, and outperforms DTI, which is particularly unreliable at finer spatial scales and tends to underestimate cortico-cortical connectivity (Calabrese et al., 2015b). Moreover, DTI results critically depend on the choice of parameters in the tractography algorithm and suffer from the accumulation of errors during the algorithmic detection of connections, in particular at fiber crossings (Thomas et al., 2014). The connectivity could be further improved by integrating experimental insights such as the dual counterstream organization of feedback and feedforward projections (Markov et al., 2014b) or different numbers of synapses per sending neuron in the feedback or feedforward connections (Rockland, 2004).

The areas of the models share the microcircuit model of Potjans & Diesmann (2014) as a common prototype for their internal connections, which we customize to the areas by taking into account different cell densities and laminar thicknesses. Combining data on cell densities with the categorization of areas into architectural types, the model features a trend of decreasing neuron numbers from early visual areas to higher hierarchical areas. This is in line with a recent report by Charvet et al. (2015), who found neuron numbers to diminish from caudal areas, at the lower end of the visual hierarchy, to rostral areas at the top of the visual hierarchy. To determine overall synapse numbers, we assumed a constant synaptic volume density (Cragg, 1967; O'Kusky & Colonnier, 1982). Combined with the neuron numbers, this leads to higher synapse numbers per neuron in higher areas, in conformity with studies that found an increase in spine numbers per pyramidal neuron from early visual areas to prefrontal cortex (Elston & Rosa, 2000; Elston, 2000; Elston et al., 2011).

The total thickness of an area and the relative thickness of its layer 4 decrease systemati-

cally with its overall cell density, with V1 having the lowest total thickness. Wagstyl et al. (2015) found thicknesses, obtained from MRI measurements, to decrease with architectural type. We also tested for a dependency with architectural types and found these fits to be less significant. From the micrographs, we determined relative instead of absolute thicknesses of layers, which circumvents the potential problem of uniform shrinkage of the tissue and obliqueness of sections. We used representative positions in the micrographs instead of systematically averaging across the graph. Hilgetag & Barbas (2006) reported a correlation between laminar thicknesses and the gyral and sulcal locations of areas, but our data did not show such trends (Fig. 6.1). In future, it may be possible to include more complete datasets on cortical thicknesses gained from MRI measurements (Calabrese et al., 2015a) or anatomical studies. However, if one combines neuronal densities and laminar thicknesses to compute neuron numbers, the effects of shrinkage tissue approximately cancel out so that using data from histological datasets for both modalities leads to a more consistent result. This can be further improved by taking into account that determining neuronal densities suffers from shrinkage in multiple dimensions while thicknesses are measured in only one dimension.

Despite these customizations to the individual areas, taking the microcircuit model as a prototype for all areas is still a drastic simplification. In future, this could be improved, e.g., by integrating the specific data on spine densities from Elston & Rosa (2000); Elston (2000); Elston et al. (2011) or by using a recent study suggesting a microcircuit for agranular cortex (Beul & Hilgetag, 2015), in our model represented by area TH.

We distinguish between synapse and cell-body patterns on the target side, using data of Binzegger et al. (2004), which renders feedback and feedforward patterns more similar in terms of soma locations than synaptic locations, in line with findings from early visual areas of mouse cortex (De Pasquale & Sherman, 2011). To distribute cortico-cortical synapses among target and source layers, we take into account connection-specific data from Markov et al. (2014b) and CoCoMac. This leads to complex target patterns, confirming that simplified schemes on connection patterns have to be relaxed (cf. Sec. 1.2).

Integration of dynamical constraints by analytical theory

In Chap. 4, we analyze the stationary fixed points of the constructed model with an analytical theory and find a bistable phase space of the model with two stable attractors. While the high-activity attractor generally yields unrealistically high rates, the low-activity attractor features low spike rates with vanishing activity in most excitatory populations of layers 5 and 6. We therefore developed a theoretical method to gain insights into the structure-dynamics relationship of the model. Weak constraints on the activity gained from electro-physiological

experiments on spiking rates to be in the range of $0.1 - 30$ spikes/s lead to systematic modifications of the model connectivity, with only small relative changes in total indegrees of each population. These modifications prove to be sufficient to raise the level of excitation in the model. As mentioned, the internal circuitry of the areas is based on the microcircuit model of Potjans & Diesmann (2014) and adapted to area-specific cell densities and laminar thicknesses. Areas with similar architecture have similar connectivity; a simplification that is softened by our method by rendering the internal connectivity more heterogeneous. The changes in the different areas can be categorized in multiple clusters, showing that we effectively refine the purely anatomy-based categorization with dynamical information from experiments.

To define the cortico-cortical connectivity in the model, we use information from a multitude of tracing studies collected in the CoCoMac database (Stephan et al., 2001). We here follow the simple approach of accumulating all information, i.e., we include a connection in the model if at least one study has reported it. Alternatively, one could consider how consistently a connection is reported across different studies (Schmitt et al., 2014). Our method removes cortico-cortical connections from the model. Compared to the entirety of connections, these links have been more frequently reported by only one study, showing that our method effectively sorts out uncertain connections.

Connection densities are generally reduced, with the most significant change in the loop between frontal areas 46 and FEF. This loop proves to be critical for global stability and our method suggests a weaker coupling than found in the structural experiment of Markov et al. (2014a). This is a possible interpretation in the light of the given uncertainties which are partly introduced by mapping between the different cortical parcellation schemes. Both areas are part of prefrontal cortex and interconnected with other cortical regions which are not included in the model. These connections might play a stabilizing role for 46 and FEF. An experimental study or an extended model could test both explanations.

Besides the densities of connections between areas, their laminar patterns are modified as well, with the strongest changes occurring for weak connections. This generally weakens the hierarchical relations in the model, possibly a reason why the model in its current state does not reproduce laminar-specific interaction patterns between areas (van Kerkoerle et al., 2014; Bastos et al., 2015a). To preserve such relations, it may be necessary to include further dynamical constraints into the model definition which will probably not yield a simple scheme, given the complex known synaptic patterns (Felleman & Van Essen, 1991; Krumnack et al., 2010; Bakker et al., 2012). Conversely, this shows that such distinct laminar patterns are not necessary to produce plausible laminar patterns of firing rates in a reasonable range.

We found that excitatory cells in layer 5 play a key role for the stability of the network. Experimental studies have found this, too, and attributed it to single-cell properties, with

pyramidal cells in layer 5 exhibiting pacemaker activity (Le Bon-Jego & Yuste, 2007; Lőrincz et al., 2015; Neske et al., 2015). Our analysis uncovers the role of structural connectivity for this behavior and suggests that it is jointly produced by both mechanisms.

The theory reliably predicts the location of the three fixed points of the systems. It relies on the diffusion approximation, which is valid for high indegrees and low synaptic weights, both fulfilled in the multi-area model. The spiking activity in the low-activity attractor is well captured as a Poisson process, explaining the good agreement between theory and experiment. The high-activity attractor generally violates the diffusion approximation used to derive the expression for the input fluctuations. Since the activity of the neuron is driven by high mean input (similar to the situation in Fig. 1.6C), input fluctuations do not play a role for the output rate of the neurons and the theory thus provides a feasible approximation. The unstable fixed point has predominantly low rates in most populations and high rates in some critical populations. This state cannot be stably reached in simulations but because it appears to be a mixture of states that are each well captured by the theory, the location of the unstable fixed point is presumably predicted with sufficient accuracy.

To preserve the global stability of the network, we keep the position of the separatrix invariant, which we achieve by controlling the position of the unstable fixed points, the only known points on the separatrix. To this end, we use a linearization around the unstable fixed point, which works well for the multi-area model because the network operates close to instability and therefore requires only small parameter changes. Larger parameter modifications can be achieved by iterative application with small parameter changes.

In the model, we use leaky integrate-and-fire model neurons with current-based synapses, which are analytically tractable. This reduces the complexity on the single-cell level and brings out the influence of the connectivity structure. The method can be extended to more complex neuron models by replacing the Siegert formula by the equivalent expression for the neuron model, be it an analytical expression or an interpolated function measured in a numerical simulation. For instance, one could model synaptic inputs as changes in conductance. Since conductance-based integrate-and-fire neurons exhibit a non-monotonic gain function with decreasing gain if the inputs are balanced between excitation and inhibition (Kuhn et al., 2004), this can render the system more stable. In our model, however, the instability is caused by excitatory inputs and conductance-based neurons show a monotonic gain function in dependence of excitatory inputs only. Therefore we would expect to observe the bistability in this case as well.

In biological neural networks, mechanisms such as short-term plasticity (reviewed in Morrison et al., 2008), spike-frequency adaptation (summarized in Benda & Herz, 2003) and homeostatic synaptic scaling (Turrigiano, 2008) increase system stability. Introducing such mecha-

nisms into a network model increases its complexity and makes an the understanding of the structure-dynamics relationship more difficult. Since we are interested in exposing the influence of structure on dynamics, we start with this mean-field description of the model, ignoring such details.

In our study, we have focused on the modification of synaptic indegrees. However, the method can be generalized to any model parameter. Given the incompleteness of experimental information, another natural choice would be synaptic weights. Our method could give hints on plasticity rules acting on the weights of single synapses, which dynamically stabilize the model.

Since experimental data on stationary activity is sparse, we have here defined only weak constraints on the dynamics. The resulting activity is therefore an emergent property of the model. With more detailed experimental insights, it would also be possible to predefine a more specific desirable state and investigate the necessary changes to achieve it.

The modifications we found and applied to the model form the basis for the simulations described in Chap. 5.

Dynamical results of the spiking network model

The network exhibits a stable resting state with asynchronous, irregular spiking activity and population bursts mediating inter-area interactions. Laminar firing rate patterns are heterogeneous across areas. In line with experimental findings (Swadlow, 1988; Fujisawa et al., 2008; Sakata & Harris, 2009), inhibitory neurons principally emit more spikes per second than excitatory cells, though this behavior is partly masked by the higher involvement of excitatory populations in cortico-cortical interactions. With all cells having identical parameters, we can ascribe this effect to the network connectivity.

Inter-area interactions trigger bursty spiking and thus longer intrisic time scales in higher visual areas. Though the model does not feature a continuous increase of time scales along the visual hierarchy, as suggested by Murray et al. (2014), it reproduces the systematic difference between MT and LIP and time scales are in the same order of magnitude ($\sim$ 100 ms) of the experimental data. These long time scales are caused by increased indegrees in higher areas and long-range interactions, making them a multi-scale feature of the network. In cortex, synaptic properties such as heterogeneous receptor densities across areas and layers (Zilles et al., 2004; Hawrylycz et al., 2012) may have an impact on the time scales of neuronal spiking.

The interactions between areas are mediated by population bursts with different degrees of synchrony. In the brain, the degree of synchrony in inter-area interactions is unclear. It may be possible to reduce the synchrony in the model by introducing nonrandom connectivity and reducing noise. Theoretical studies have shown that high impact of noise on neurons can reduce

the sensitivity of neurons to external perturbations and thus decrease the effective coupling in the network (Aertsen & Preißl, 1990). One possible mechanism of the brain to counteract this may be to adjust the external noise to the intrinsic connectivity (Marre et al., 2009). Enhanced intrinsic interactions in a stable network may be achieved in a model by adding nonrandom structures such as embedded feedforward networks, e.g. synfire chains (Diesmann et al., 1999), small-world connectivity (Jahnke et al., 2014), population-specific short-term plasticity mechanisms (Sussillo et al., 2007) or fine-tuned inhibition between groups of neurons (Hennequin et al., 2014).

Population bursts propagate stably across the network predominantly in feedback direction, with the exception of frontal areas 46 and FEF which are placed at the top of the hierarchy due to their anti-correlation with the early visual areas (Fig. 6.6). Similarly, recordings of electroencephalograms (EEG) have shown that slow waves during sleep (Massimini et al., 2004; Nir et al., 2011; Sheroziya & Timofeev, 2014) and information flow during visual imagery (Dentico et al., 2014) propagate in feedback direction from parietal to occipital areas. The employed procedure is similar to a method used for fMRI recordings (Mitra et al., 2014) and could be extended to distinguish between excitatory and inhibitory influences between areas like the observed interactions between early visual and frontal areas. Inter-area projections predominantly target excitatory cells. Therefore, the increase of cortico-cortical synaptic weights to enhance coupling between areas has to be accompanied by a further increase of the synaptic weights of connections onto inhibitory cells to maintain network stability. This is reminiscent of the *handshake effect* in the microcircuit model (Potjans & Diesmann, 2014) where the network is stabilized due to the inhibitory net effect of interlaminar projections.

As a sanity check for the structural connectivity, we partition the resulting directed and weighted graph into six clusters. They reflect knwon anatomical and functional similarities of the areas by grouping the areas into early visual and frontal areas as well as dorsal and ventral stream. Moreover, the graded structural connectivity revealed a better agreement with the experimental resting-state data than the binary adjacency matrix. This underlines the importance of heterogeneous connection strengths and the plausibility of the structural connectivity.

The cortico-cortical interaction patterns in the model agree well with experimental observations and exceed the correlation of the structural matrix with the experiment, proving that the dynamical simulation adds information to the structural connectivity. The most obvious discrepancies occur for MDP and the frontal areas. While MDP is uncoupled from the rest of the model, the experimental data reveal significant interactions, suggesting that the lack of incoming connections is a deficiency of the dataset. The frontal areas 46 and FEF are significantly connected to non-visual areas, which could lead to indirect coupling to visual areas, not captured in our purely visual model. On the other hand, the model reproduces the anti-

correlation between early visual and frontal areas, which is an interesting feature in regards of cortical function because these two groups of areas fulfil complementary tasks at both ends of visual processing.

In conclusion, the presented network model combines important features of cortical activity from single-cell spiking to large-scale interactions made possible by the multi-scale approach we have taken in this study.

Future work and outlook

This work is part of continuous efforts. The following section sketches an outlook on possible future projects which arise from our work.

Analyzing the memory consumption of NEST shows that, with growing machine sizes, the sparse table, holding 1 bit for each neuron in the network on each thread, will ultimately limit the scalability of the software. The sparse table is necessary because NEST uses a collective communication scheme for spikes, i.e., all spikes are communicated to all virtual processes which then distribute them to the postsynaptic neurons. For current machine sizes, this is a feasible approach because with approximately 2,000 neurons per core each receiving 10,000 synapses, the probability that at least one spike has to be communicated between any pair of cores is close to one. This probability decreases with increasing numbers of cores and thus directed communication between threads will become favorable. This will supersede the sparse table and save a significant amount of memory overhead. On the other hand, it requires two structures to be implemented. First, each neuron needs to know its postsynaptic partners so that the software knows where to send the emitted spikes. Second, each receiving core needs a map from the ID of the sending neuron to its target partners. Therefore, future work needs to find a memory-efficient structure and fast algorithms for the implementation of these structures.

The multi-area model of macaque visual cortex is a first step towards a self-consistent spiking network model of the cortex and can act as a basis for more refined models. In its current state, the model does not include any local topography, although cortical areas are known to have spatially dependent connectivity (for instance reviewed in Voges et al., 2010) and cortico-cortical connections exhibit a certain degree of convergence and divergence (Colby et al., 1988; Salin et al., 1989; Gattass et al., 1997; Markov et al., 2014b). Interactions in the model currently occur on fast time scales on the order of the inter-areal delays ($\sim$ 10 ms), while cortical activity is known to exhibit slow oscillations on the order of 100 ms to seconds measured in spiking and LFP activity (Belitski et al., 2008) as well as in fMRI experiments (Vincent et al., 2007; Shen et al., 2012, 2015). Such long time scales can be caused by various mechanisms. On the one hand, it is known that neuronal connections intrinsically have long time scales in

NMDA synapses (Zito & Scheuss, 2009). On the other hand, the network can exhibit slow fluctuations if the eigenmodes of the effective connectivity are close to instability (Bos et al., 2015) or non-orthogonal eigendirections couple (Hennequin et al., 2012). Both effects can be tested and separated from each other with the multi-scale model we have developed in this study.

With the analytical theory, we have integrated dynamical constraints into the model definition. So far, this is restricted to first-order statistical measures of spiking activity, i.e., population rates. In future, the method can be extended to include second-order statistics, such as power spectra and correlations of spiking activity. For instance, fMRI measurements of resting states (Shen et al., 2012, 2015) inform about interactions between areas. In Chap. 5, we demonstrate significant agreement between the simulated and experimentally found functional connectivity. However, the interactions between clusters show larger deviations than within clusters. Constraining the model with the experimentally found functional connectivity could improve the agreement and shed further light on the structural patterns underlying inter-area communication.

IV

Appendices

CHAPTER 6

Supplementary material

This chapter gives supplementary information to the main text of this thesis, including detailed derivations for the leaky integrate-and-fire neuron, definitions of all dynamical measures used throughout the thesis and additional information to the results section.

6.1 Leaky integrate-and-fire neuron

The membrane potential V and the synaptic input I_s of leaky integrate-and-fire neurons with exponentially decaying postsynaptic currents are described by two differential equations reading

$$\begin{aligned} \tau_\mathrm{m}\frac{\mathrm{d}V}{\mathrm{d}t} &= -(V-E_\mathrm{L}) + \frac{\tau_\mathrm{m}}{C_\mathrm{m}} I_\mathrm{s}(t) \\ \tau_\mathrm{s}\frac{\mathrm{d}I_\mathrm{s}}{\mathrm{d}t} &= -I_\mathrm{s} + \sum_i \delta(t-t_i)J\,, \end{aligned} \tag{6.1}$$

A single spike at time t_1

$$\tau_\mathrm{s}\frac{\mathrm{d}I_\mathrm{s}}{\mathrm{d}t} = -I_\mathrm{s} + \delta(t-t_1)J \tag{6.2}$$

leads to a **post-synaptic current**

$$\begin{aligned} PSC(t) &= C(t)\mathrm{e}^{-t/\tau_\mathrm{s}} \\ \text{with } C(t) &= J\mathrm{e}^{t_1/\tau_\mathrm{s}}\Theta(t_1) \\ \Rightarrow PSC(t) &= J\mathrm{e}^{-(t-t_1)/\tau_\mathrm{s}}\Theta(t_1) \end{aligned} \tag{6.3}$$

which evokes a **post-synaptic potential**

$$\begin{aligned} \tau_\mathrm{m}\frac{\mathrm{d}PSP}{\mathrm{d}t} &= -PSP + \frac{\tau_\mathrm{m}}{C_\mathrm{m}} J\mathrm{e}^{-(t-t_1)/\tau_\mathrm{s}}\Theta(t_1) \\ \Rightarrow PSP(t) &= C_2(t)\mathrm{e}^{-t/\tau_\mathrm{m}} \end{aligned} \tag{6.4}$$

with

$$\begin{aligned} \frac{\mathrm{d}C_2(t)}{\mathrm{d}t}; &= \frac{1}{C_\mathrm{m}} J\mathrm{e}^{t_1/\tau_\mathrm{s}}\mathrm{e}^{-t\left(\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}\right)}\Theta(t_1) \\ \Rightarrow C_2(t) &= \frac{1}{C_\mathrm{m}}\frac{J\mathrm{e}^{t_1/\tau_\mathrm{s}}}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\left[\mathrm{e}^{-t_1\left(\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}\right)} - \mathrm{e}^{-t\left(\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}\right)}\right] \\ \Rightarrow PSP(t) &= \frac{1}{C_\mathrm{m}}\frac{J}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\left[\mathrm{e}^{-\frac{t-t_1}{\tau_\mathrm{m}}} - \mathrm{e}^{-\frac{t-t_1}{\tau_\mathrm{s}}}\right]\Theta(t_1)\,. \end{aligned} \tag{6.5}$$

The **integral of the** PSP reads

$$\begin{aligned} \int_0^\infty PSP(t)\mathrm{d}t &= \frac{1}{C_\mathrm{m}}\frac{J}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\int_{t_1}^\infty \left[\mathrm{e}^{-\frac{t-t_1}{\tau_\mathrm{m}}} - \mathrm{e}^{-\frac{t-t_1}{\tau_\mathrm{s}}}\right]\mathrm{d}t \\ &= \frac{1}{C_\mathrm{m}}\frac{J}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\left[\tau_\mathrm{m}-\tau_\mathrm{s}\right]\mathrm{d}t \\ &= \frac{J}{C_\mathrm{m}}\tau_\mathrm{m}\tau_\mathrm{s}\,, \end{aligned} \tag{6.6}$$

and the **maximum of the** PSP amounts to

$$\begin{aligned}\frac{\mathrm{d}PSP}{\mathrm{d}t} &= \frac{-1}{C_\mathrm{m}}\frac{J}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\left[\frac{1}{\tau_\mathrm{m}}\mathrm{e}^{-\frac{t-t_1}{\tau_\mathrm{m}}}-\frac{1}{\tau_\mathrm{s}}\mathrm{e}^{-\frac{t-t_1}{\tau_\mathrm{s}}}\right]\stackrel{!}{=}0\\ &\Leftrightarrow t-t_1=\log\left(\frac{\tau_\mathrm{m}}{\tau_\mathrm{s}}\right)\frac{1}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\\ \Rightarrow PSP_\mathrm{max} &= \frac{J}{C_\mathrm{m}}\frac{1}{\frac{1}{\tau_\mathrm{s}}-\frac{1}{\tau_\mathrm{m}}}\left[\left(\frac{\tau_\mathrm{m}}{\tau_\mathrm{s}}\right)^{-\frac{1}{1-\frac{\tau_\mathrm{s}}{\tau_\mathrm{m}}}}-\left(\frac{\tau_\mathrm{m}}{\tau_\mathrm{s}}\right)^{-\frac{1}{\frac{\tau_\mathrm{m}}{\tau_\mathrm{s}}-1}}\right].\end{aligned} \tag{6.7}$$

For a LIF neuron driven by a constant current μ, (6.1) becomes

$$\tau_\mathrm{m}\frac{\mathrm{d}V}{\mathrm{d}t}=-(V-E_\mathrm{L})+\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\mu\,, \tag{6.8}$$

which is solved by

$$V(t)=\left(E_\mathrm{L}+\mu\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\right)\left(1-\mathrm{e}^{-\frac{t}{\tau_\mathrm{m}}}\right). \tag{6.9}$$

The membrane potential thus settles in

$$\lim_{t\to\infty}V(t)=E_\mathrm{L}+\mu\frac{\tau_\mathrm{m}}{C_\mathrm{m}}\,, \tag{6.10}$$

so that we can define the current μ_th necessary to drive the membrane potential exactly to the threshold to

$$\mu_\mathrm{th}=(\theta-E_\mathrm{L})\,\frac{C_\mathrm{m}}{\tau_\mathrm{m}}\,. \tag{6.11}$$

Furthermore, we define the asymptotic free membrane potential V_∞ as the asymptotic membrane potential when setting $\theta\to\infty$, i.e., the membrane potential can freely develop without being reset when crossing the threshold.

6.1.1 Campbell's theorem

An irregular spiking neuron can be approximated as Poisson point process with the parameter set to the firing rate ν. A connected neuron then receives spikes generated from a Poisson point process $z(t)=\sum_i\delta(t-t_i)=\sum_i\delta_i$ and convolves them with the exponential post-synaptic current $PSC(t)$, thus the input current to the neuron reads

$$I_\mathrm{s}(t)=\sum_i\left[PSC*\delta_i\right](t) \tag{6.12}$$

Campbell's theorem (Papoulis & Pillai, 2002) then states that the expectation value of the input current is

$$\mathbb{E}\left(I_\mathrm{s}(t)\right)=\nu\int_{-\infty}^{\infty}PSC(t)\mathrm{d}t\,, \tag{6.13}$$

and its variance

$$\mathrm{Var}\left(I_\mathrm{s}(t)\right) = \nu \int_{-\infty}^{\infty} PSC^2(t)\mathrm{d}t\,. \tag{6.14}$$

6.1.2 Derivation of the Siegert formula

In this section, we will first derive the Fokker-Planck equation describing the time evolution of the probability density function for the state of the neuron, and then deduce the Sieger equation as the stationary solution of the Fokker-Planck equation. The derivation is based on Risken (1996, p.65, "third way") and follows the script of the lecture "Theoretical Neuroscience: Correlation structure of neuronal networks" held from 2013 to 2015 at RWTH Aachen University.

We start with the basic equation for the leaky integrate-and-fire neuron (cf. (1.1)), first consider the simpler case of a δ-shaped synaptic inputs, i.e.,

$$\begin{aligned} \tau_\mathrm{m}\frac{\mathrm{d}V}{\mathrm{d}t} &= -(V - E_\mathrm{L}) + \frac{\tau_\mathrm{m}}{C_\mathrm{m}} I_\mathrm{s}(t) \\ \tau_\mathrm{s} I_\mathrm{s}(t) &= \sum_i J\delta(t - t_i) \end{aligned} \tag{6.15}$$

and later generalize this to exponential synaptic input currents as used throughout the main text of this thesis. In the context of a balanced random network, spikes are usually emitted irregularly so that we assume the input spike trains as a Poisson process. This assumption renders (6.15) a stochastic difffential equation and the membrane potential dynamics is a Markov process because the neuron's state is completely defined by the membrane potential and the history of the membrane potential does not matter. We are not interested in solving (6.15) for a particular instance of the stochastic input spike train, but want to determine the probability density $\rho(y,t)$ for the neuron's membrane potential to be of value y at time t, defined as

$$\rho(y,t) = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathcal{P}(V(t)\in[y, y+\epsilon]) \tag{6.16}$$

As a probability density, ρ has to be normalized as

$$\int_{-\infty}^{\infty}\rho(y,t)\mathrm{d}y = 1\,\forall t\,. \tag{6.17}$$

Since the time evolution of the neuron is a Markov process, the probability of the neuron to be in the interval $[y, y+\epsilon]$ at time t given that it had the precise value x at time s is

$$\mathcal{P}(y,t|x,s) = \lim_{\epsilon\to 0}\frac{1}{\epsilon}\mathcal{P}(V(t)\in[y,y+\epsilon]|V(s)=x)\,, \tag{6.18}$$

because this probability does not depend on the history prior to s. Summing over all possible target states, the probability must be one, i.e.,

$$\int_{-\infty}^{\infty} \mathcal{P}(y,t|x,s)\mathrm{d}y = 1 \,\forall x,t,s\,. \tag{6.19}$$

Using $\mathcal{P}(y) = \int_{-\infty}^{\infty} \mathcal{P}(y|x)\mathcal{P}(x)$, we obtain the **Chapman-Kolmogorov equation** for $\rho(y,t)$, which reads for $t > s$

$$\rho(y,t) = \int_{-\infty}^{\infty} \mathcal{P}(y,t|x,s)\rho(x,s)\mathrm{d}x\,. \tag{6.20}$$

In the limit $t \to s$, we observe that

$$\lim_{t\to s} \mathcal{P}(y,t|x,s) = \delta(x-y)\,. \tag{6.21}$$

As a next step, we derive a differential formulation of (6.20), the **Fokker-Planck equation**. It relies on the assumption that the time evolution of the membrane potential only makes small jumps $\Delta = x - y$, which is equivalent to assuming small synaptic weights J. For small δt, $\mathcal{P}(y, t+\delta t|x,t)$ must have a maximum neat $x \approx y$ because it smoothly approaches $\delta(x-y)$ for $\delta t \to 0$. The idea now is to describe this maximum as a parabola, i.e., to approximate the dependence of P as a Taylor expansion up to second order. Finally, we want to obtain an expression with derivatives in y. We can transform all derivatives ∂_Δ into $-\partial_y$ if there are only terms in which Δ appears with y as a difference. To achieve this, we introduce the auxiliary function

$$f(\Delta,\Delta') = \mathcal{P}(y+\Delta'-\Delta, t+\delta t|y-\Delta,t)\rho(y-\Delta,t)\,, \tag{6.22}$$

which we expand to 2nd order

$$\begin{aligned} f(\Delta,\Delta') &= (1 + \Delta\partial_\Delta + \frac{1}{2}\Delta^2\partial_\Delta^2) f(\Delta',\Delta)|_{\Delta=0} + \mathcal{O}(\Delta^3) \\ &= (1 - \Delta\partial_y + \frac{1}{2}\Delta^2\partial_y^2)\mathcal{P}(y+\Delta', t+\delta t|y,t)\rho(y,t)\,, \end{aligned} \tag{6.23}$$

leading to

$$f(\Delta,\Delta) = \mathcal{P}(y,t+\delta t|y-\Delta,t)\rho(y-\Delta,t) = (1-\Delta\partial_y+\frac{1}{2}\Delta^2\partial_y^2)\mathcal{P}(y+\Delta,t+\delta t|y,t)\rho(y,t)\,. \tag{6.24}$$

This gives an approximation of the right-hand side of (6.20),

$$\begin{aligned} \rho(y,t+\delta t) &= \int_{-\infty}^{\infty} \mathcal{P}(y,t+\delta t|y-\Delta,t)\rho(y-\Delta,t)\mathrm{d}\Delta \\ &= \int_{-\infty}^{\infty} \left(1 - \Delta\partial_y + \frac{1}{2}\Delta^2\partial_y^2\right)\mathcal{P}(y+\Delta,t+\delta t|y,t)\rho(y,t)\mathrm{d}\Delta\,. \end{aligned} \tag{6.25}$$

Defining

$$a_n(y, t+\delta t, t) := \int_{-\infty}^{\infty} \Delta^n \mathcal{P}(y+\Delta, t+\delta t|y, t)\mathrm{d}\Delta \tag{6.26}$$

we can rewrite (6.25) as

$$\rho(y, t+\delta t) - \rho(y, t) = \left(-\partial_y a_1(y, t+\delta t, t) + \frac{1}{2}\partial_y^2 a_2(y, t+\delta t, t)\right)\rho(y, t) \tag{6.27}$$

where the derivates also act on $\rho(y, t)$. The interpretation of the a_i is:

- a_1 describes the mean drift of the trajectories starting at t at point y after a short interval δt.

- a_2 describes the spread of the trajectories in the short interval δt.

Taking the limit $\delta t \to 0$ and introducing

$$A_n(y, t) = \lim_{\delta t \to t} \frac{1}{\delta t} a_n(y, t+\delta t, t)\,, \tag{6.28}$$

we obtain the Fokker-Plack equation

$$\begin{aligned} \partial_t \rho(y, t) &= \left(-\partial_y A_1(y, t+\delta t, t) + \frac{1}{2}\partial_y^2 A_2(y, t+\delta t, t)\right)\rho(y, t) \\ &= -\partial_y \Phi(y, t)\,, \end{aligned} \tag{6.29}$$

where we introduced the probability flux Φ, showing that (6.29) is a continuity equation.
We now calculate the moments a_i for a LIF neuron receiving spikes from two Poisson processes, one sending excitatory spikes with rate ν_e with amplitude J and one sending inhibitory spikes with rate ν_i with rate ν_i. Two different scenarios occur after an infinitesimal time step δt:

1. **No incoming spike:** The membrane potential drops by $\Delta = V(t+\delta t) - V(t) = V\mathrm{e}^{-\frac{\delta t}{\tau_\mathrm{m}}} - V = V\left(\mathrm{e}^{-\frac{\delta t}{\tau_\mathrm{m}}} - 1\right) = -\frac{\delta t}{\tau_\mathrm{m}} + \mathcal{O}(\delta t^2)$. This happens with probability $1 - (\nu_\mathrm{e} + \nu_\mathrm{i})\delta t$.

2. **Spike of type** $x \in \{\mathrm{e}, \mathrm{i}\}$: The membrane potential evolves with a jump so that $\Delta = J_x + V\left(\mathrm{e}^{-\frac{\delta t}{\tau_\mathrm{m}}} - 1\right) = J_x - \frac{\delta t}{\tau_\mathrm{m}} + \mathcal{O}(\delta t^2)$. This happens with probability $\nu_x \delta t$.

We drop all terms of order δt^2 or higher and reduce the integral in (6.26) to a sum over the two scenarios

$$a_n(V, t+\delta t, t) = \sum_{x \in \{\mathrm{e},\mathrm{i}\}} \left(J_x - \frac{\delta t}{\tau_\mathrm{m}} V\right)^n \nu_x \delta t + \left(\frac{-\delta t}{\tau_\mathrm{m}} V\right)^n (1 - (\nu_\mathrm{e} + \nu_\mathrm{i}\delta t)) \tag{6.30}$$

For $n = 1$, we get

$$\begin{aligned} a_1(V, t+\delta t, t) &= \sum_{x\in\{\mathrm{e,i}\}} J_x \nu_x \delta t - \frac{\delta t}{\tau_\mathrm{m}} V + \mathcal{O}(\delta t^2) \\ A_1(V,t) &= \lim_{\delta t\to 0} \frac{1}{\delta t} a_1(V, t+\delta t, t) \\ &= \underbrace{\sum_{x\in\{\mathrm{e,i}\}} J_x \nu_x}_{=:\mu/\tau_\mathrm{m}} - \frac{1}{\tau_\mathrm{m}} V \\ &= -\frac{1}{\tau_\mathrm{m}} (V - \mu) \end{aligned} \tag{6.31}$$

For $n = 2$, we get

$$\begin{aligned} a_1(V, t+\delta t, t) &= \sum_{x\in\{\mathrm{e,i}\}} J_x^2 \nu_x \delta t + \mathcal{O}(\delta t^2) \\ A_2(V,t) &= \sum_{x\in\{\mathrm{e,i}\}} J_x^2 \nu_x =: \frac{\sigma^2}{\tau_\mathrm{m}} \end{aligned} \tag{6.32}$$

The **Fokker-Planck equation** thus takes the form

$$\begin{aligned} &\partial_t \rho(V,t) = -\partial_V \Phi(V,t) \text{ with} \\ &\Phi(V,t) = \frac{1}{\tau_\mathrm{m}} \left(\underbrace{V-\mu}_{\text{drift term}} - \underbrace{\frac{1}{2}\sigma^2 \partial_V}_{\text{diffusion term}} \right) \rho(V,t) \end{aligned} \tag{6.33}$$

For convenience, we transform time and voltage by introducing $s = \frac{t}{\tau_\mathrm{m}}$ and $x = \frac{\sqrt{2}}{\sigma}(V - \mu)$ and rescale ρ such that $\rho(x, s) = \frac{\sigma}{\sqrt{2}} \rho(V, t)$ is still normalized. This leads to a simpler expression for (6.33), namely

$$\begin{aligned} \partial_s \rho(x,s) &= -\partial_x \Phi(x,s) \\ \Phi(x,s) &= -\left(x + \partial_x\right) \rho(x,s) \end{aligned} \tag{6.34}$$

The drift term describes the deterministic drift of trajectories with speed $-\frac{1}{\tau_\mathrm{m}}(V - \mu)$ and the diffusion term describes the 'hopping' between neighboring states. As a last step, we determine the stationary solution of (6.33), i.e., where the probability distribution does not change over time, $\partial_s \rho(x, s) = 0$, so that the probability flux must be constant along x

$$\Phi(x,s) = \text{const.} \tag{6.35}$$

To take the threshold of the LIF neuron into account (cf. Sec. 1.4), we specify the boundary

conditions:

1. The density must vanish above threshold, because $V(t) < \theta \,\forall t$, i.e., $\rho = 0$ for $x > x_{\mathrm{th}}$.

2. After reaching threshold, the membrane potential x is reset to the reset potential x_{t}, so the flux at threshold $\Phi(x_{\mathrm{th}})$ must be re-inserted at reset, resulting in a step of the flux at reset

$$\Phi(x_{\mathrm{r}}+) - \Phi(x_{\mathrm{r}}-) = \Phi(x_{\mathrm{th}}) = \tau_{\mathrm{m}}\nu\,, \tag{6.36}$$

   where we used that the probability flux over threshold must be equal to the firing ν of the neuron.

The piecewise constant flux (6.35) therefore makes a step from 0 to $\tau_{\mathrm{m}}\nu$ at reset and back to 0 at threshold, so we need to solve the inhomogeneous linear first order differential equation (using (6.33)),

$$(x + \partial_x)\,\rho(x) = -\tau_{\mathrm{m}}\nu\Theta(x - x_{\mathrm{r}})\Theta(x_{\mathrm{th}} - x) \tag{6.37}$$

where $\Theta(x)$ denotes the Heaviside step function. The homogeneous solution reads

$$\begin{aligned} (x + \partial_x)\,\rho_{\mathrm{h}}(x) &= 0 \\ \rho_{\mathrm{h}}(x) &= \mathrm{e}^{-\frac{1}{2}x^2} \end{aligned} \tag{6.38}$$

and the particular solution, vanishing at threshold

$$\begin{aligned} \rho(x) &= \tau_{\mathrm{m}}\nu\rho_{\mathrm{h}}(x)\int_x^{x_{\mathrm{th}}} \rho_{\mathrm{h}}(x')^{-1}\Theta(x' - x_{\mathrm{r}})\mathrm{d}x' \\ &= \tau_{\mathrm{m}}\nu\mathrm{e}^{-\frac{1}{2}x^2}\int_{\max(x,x_{\mathrm{r}})}^{x_{\mathrm{th}}} \mathrm{e}^{\frac{1}{2}x'^2}\mathrm{d}x' \end{aligned} \tag{6.39}$$

We use the normalization condition to determine the firing ν and obtain

$$\begin{aligned} 1 &= \int_{-\infty}^{x_{\mathrm{th}}} \rho(x)\mathrm{d}x \\ &= \tau_{\mathrm{m}}\nu\int_{-\infty}^{x_{\mathrm{th}}} \mathrm{e}^{-\frac{1}{2}x^2}\int_{\max(x,x_{\mathrm{r}})}^{x_{\mathrm{th}}} \mathrm{e}^{\frac{1}{2}x'^2}\mathrm{d}x'\mathrm{d}x \\ &= 2\tau_{\mathrm{m}}\nu\int_{-\infty}^{x_{\mathrm{th}}/\sqrt{2}} \underbrace{\mathrm{e}^{-x^2}}_{u'}\underbrace{\int_{\max(x,x_{\mathrm{r}})}^{x_{\mathrm{th}}/\sqrt{2}} \mathrm{e}^{x'^2}\mathrm{d}x'}_{v}\,\mathrm{d}x \end{aligned} \tag{6.40}$$

which, using partial integration, $u = \frac{\sqrt{\pi}}{2}(1 + \mathrm{erf}(x))$, and $v' = -\mathrm{e}^{x^2}\Theta(x - x_\mathrm{r})$, we write as

$$\begin{aligned}(\tau_\mathrm{m}\nu)^{-1} &= \sqrt{\pi}\underbrace{(1+\mathrm{erf}(x))}_{=0 \text{ at } x=-\infty}\underbrace{\int_{\max(x,x_\mathrm{r})}^{x_\mathrm{th}/\sqrt{2}} \mathrm{e}^{x'^2}\mathrm{d}x'}_{=0 \text{ at } x=x_\mathrm{th}/\sqrt{2}}\Big|_{-\infty}^{x_\mathrm{th}/\sqrt{2}} + \sqrt{\pi}\int_{-x_\mathrm{r}/\sqrt{2}}^{x_\mathrm{th}/\sqrt{2}}(1+\mathrm{erf}(x))\mathrm{e}^{x^2}\mathrm{d}x\,. \\ (\tau_\mathrm{m}\nu)^{-1} &= \sqrt{\pi}\int_{\frac{V_\mathrm{r}-\mu}{\sigma}}^{\frac{\theta-\mu}{\sigma}}(1+\mathrm{erf}(x))\mathrm{e}^{x^2}\mathrm{d}x\,.\end{aligned} \tag{6.41}$$

This is the Siegert formula determining the firing ν of LIF neuron with δ-shaped synaptic input currents under the given assumptions (Siegert, 1951; Brunel, 2000). For exponentially decaying post-synaptic currents with $\tau_\mathrm{s} \ll \tau_\mathrm{m}$, the limits of the integral are shifted by $\gamma\sqrt{\frac{\tau_\mathrm{s}}{\tau_\mathrm{m}}}$ (Fourcaud & Brunel, 2002; Schuecker et al., 2015a) where $\gamma = |\zeta(1/2)|/\sqrt{2}$, with ζ denoting the Riemann zeta function (Abramowitz & Stegun, 1974). A refractory period τ_r introduces a fixed period where the voltage is clamped, which adds up to the right side of (6.41), so that we finally obtain

$$(\nu)^{-1} = \tau_\mathrm{r} + \tau_\mathrm{m}\sqrt{\pi}\int_{\frac{V_\mathrm{r}-\mu_i}{\sigma_i}+\gamma\sqrt{\frac{\tau_\mathrm{s}}{\tau_\mathrm{m}}}}^{\frac{\theta-\mu_i}{\sigma_i}+\gamma\sqrt{\frac{\tau_\mathrm{s}}{\tau_\mathrm{m}}}} \mathrm{e}^{x^2}\ (1+\mathrm{erf}(x))\,\mathrm{d}x\,. \tag{6.42}$$

6.2 Dynamical measures

In this section, the dynamical measures used to characterize network activity in this work are described. All measures are computed on spike trains of single neurons

$$s_i(t) = \sum_j \delta\left(t - t_j^{(i)}\right) . \tag{6.43}$$

CV ISI

The coefficient of variation of the inter-spike intervals (CV ISI) of a spike train s_i is defined as

$$CVISI_i = \frac{\sum_i \left(I_i - \overline{I_i}\right)^2}{\overline{I_i}} , \tag{6.44}$$

where I_i is the series of invervals between consecutive spikes (ISI) in s_i and $\overline{I_i}$ denotes the arithmetic mean of the ISI.

Local variance revised (LvR)

The revised local variance (LvR) of inter-spike intervals (Shinomoto et al., 2009) is defined as

$$\begin{aligned} LvR &= \frac{3}{n-1} \sum_{i=1}^{n-1} \left(\frac{I_i - I_{i+1}}{I_i + I_{i+1}}\right) \qquad \left(1 + \frac{4\tau_\mathrm{r}}{I_i + I_{i+1}}\right) \\ &= \frac{3}{n-1} \sum_{i=1}^{n-1} \left(1 - \frac{4 I_i I_{i+1}}{(I_i + I_{i+1})^2}\right) \quad \left(1 + \frac{4\tau_\mathrm{r}}{I_i + I_{i+1}}\right) , \end{aligned} \tag{6.45}$$

where the first term compute the local variance of consecutive inter-spike intervals I_i and the last term accounts for the refractoriness of the neuron. The extreme case of a clock-like spiking neuron with perfectly regular intervals leads to $LvR = 0$, while a Poisson point process yields $LvR = 1$.

Population rate

Throughout the thesis, the population firing rate $r(t)$ is computed as the histogram of spike times with bin width Δt divided by the number of neurons N, so that

$$r(k\Delta t) = \frac{1}{N} \sum_i \int_{k\Delta t}^{(k+1)\Delta t} s_i(t) \mathrm{d}t \, . \tag{6.46}$$

Synchrony

The synchrony S_i of the spiking activity in a population i is quantified by the coefficient of variation of its population spiking rate $r_i(t)$ as

$$S_i = \frac{\langle (r_i(t) - \langle r_i(t) \rangle_t)^2 \rangle_t}{\langle r_i(t) \rangle_t} \,. \tag{6.47}$$

Covariance

The covariance $\rho_{ij}(\tau)$ at time lag τ between two time series $r_i(t), r_j(t)$ is defined as

$$\rho_{ij}(\tau) = \sum_k r_i(t_k) r_j(t_k) \,, \tag{6.48}$$

Cross-correlation coefficient of a neuronal population

Another way to assess synchrony in a neural population of size N is to compute the average zero-time lag cross-correlation coefficient $\overline{cc}$ between pairs of neurons.

$$\overline{cc} = \frac{1}{N^2} \sum_{i,j} \rho_{ij}(0) \tag{6.49}$$

Functional connectivity

The functional connectivity F_{ij} between two neuronal populations i, j is defined as the zero-time lag cross-correlation coefficient of their average synaptic input currents I_i, I_j, i.e.,

$$F_{ij} = \frac{\rho_{ij}(0)}{\sqrt{\rho_{ii}(0)\rho_{jj}(0)}} \tag{6.50}$$

where $F_{ij} \in [-1, 1]$. It is a measure for the strength of the interactions between the two populations, yet does not contain information on the direction of interactions.

6.3 Appendix to Chap. 2: "Spiking network simulation code for petascale computers"

Algorithm 1 Recursive definition of homogeneous containers implementing low-overhead vector for small numbers of elements. The K data elements (synapse objects) of template type `connectionT` are stored in a C-style array (line 8) and hence consume $K \cdot$sizeof(connectionT). 8 B of additional overhead are due to the virtual function table (vtable) pointer inherited from the abstract base class `ConnectorBase` (see also Fig. 2.4). The first template definition (line 4) serves as the recursion continuation step with the number K of stored elements as the recurrence parameter. The method `push_back` is inherited from the abstract base class `vector_like`. For template arguments $K < K_{\text{cutoff}}$ push_back recursively instantiates a container of size $K+1$ (line 27) at compile time, joins the previously stored contents with the new element and destroys itself (line 31). The new container is returned to the caller and is stored in the adaptive data structure (see also Alg. 2). Recursion termination is achieved by providing a template specialization for K_{cutoff} (line 39), which uses a `std::vector<connectionT>` to store the data elements. $K_{\text{cutoff}} = 3$ is a preprocessor define fixed at compile time.

```
1  // inheritance from abstract base class
2  // 1. vector_like: to provide push_back as common interface
3  // 2. ConnectorBase: to have common base class type for all containers
4
5  // recursion step
6  template<int K, typename connectionT>
7  struct HomConnector
8    : public vector_like<connectionT>
9  {
10     connectionT C_[K];
11
12     HomConnector(const HomConnector<K-1, connectionT> & Cm1,
13                  const connectionT & c)
14     : syn_id_(Cm1.syn_id_)
15     {
16         for (int i=0; i<K-1; i++)
17             C_[i] = Cm1.C_[i];
18         C_[K-1] = c;
19     }
20
21     // pushback: recursion step to next container
22     HomConnector<K+1, connectionT> & push_back(const connectionT & c)
23     {
24        // pass on contents and
25        // recursively instantiate container for K+1 elements
26        // use pool allocator, pass (*this, c) as args to constructor
27        HomConnector<K+1, connectionT> *newconn =
28             allocate<HomConnector<K+1, ConnectionT> >(*this, c);
29
30        // delete this instance
31        delete this;
32
33        // return container of next size
34        return *newconn;
35     }
36 };
```

```
37
38 // recursion termination by template specialization
39 template<typename connectionT>
40 struct HomConnector<K_cutoff, connectionT>
41   : public vector_like<connectionT>
42 {
43     std::vector<connectionT> C_;
44     ...
45     HomConnector<K_cutoff, connectionT> & push_back(const connectionT & c)
46     {
47         C_.push_back(c);
48         return *this;
49    }
50 };
```

Algorithm 2 Algorithm to create new connections using adaptive data containers. The algorithm is implemented in the method `add_connection()` of `ConnectorModel` (see also Fig. 2.4). The template function `allocate<T>` invokes the custom pool allocator, providing for each thread memory from a separate pool. The different branches of the algorithm marked by Case 0-3 refer to the description in the main text.

```
1  ConnectorBase* add_connection(Node& tgt, Node& src,
2                                ConnectorBase* conn, ConnectionT& c):
3
4      // source neuron does not yet have any local targets (Case 0)
5    if conn == 0:
6      c.check_connection(t_lastspike)
7      conn ← allocate<HomConnector<1, ConnectionT> >(c)
8
9    // source neuron has at least one local target
10   else:
11     c.check_connection(t_lastspike(*conn).get_t_lastspike())
12
13     // all existing local synapses are of the same type (Case 1,2)
14     if (*conn).homogeneous_model():
15
16       // new synapse is of same type as existing ones (stay in Case 1,2)
17       if (*conn).get_syn_id() == c.get_syn_id():
18         hom_vec ← static_cast<vector_like<ConnectionT> * >(conn)
19         conn ← &(*hom_vec).push_back(c)
20
21       // new synapse is of different type than existing ones (switch to Case 3)
22       else:
23         het_conn ← allocate<HetConnector>()
24         (*het_conn).push_back(conn)
25         hom_conn ← allocate<HomConnector<1, ConnectionT> >(c)
26         (*het_conn).push_back(hom_conn)
27         conn ← het_conn
28
29     // different types of local synapses exist (Case 3)
30     else:
31       het_conn ← static_cast<HetConnector>(conn)
32       foreach h_conn in het_conn:
33
34         // at least one local synapse of this type already exist
35         if (*h_conn).get_syn_id() == c.get_syn_id():
36           hom_conn ← static_cast<HomConnector<ConnectionT> * >(h_conn)
37           hom_conn ← &(*hom_conn).push_back(c)Reduced overhead of b
38           found ← true
39           break
40
41       // local synapses of this type do not yet exist
42       if not found:
43         hom_conn ← allocate<HomConnector<1, ConnectionT> >(c)
44         (*het_conn).push_back(hom_conn)
45
46   return conn
```

Algorithm 3 `get_node_by_gid()` in `SparseNodeArray` returns `0` for non-local nodes and the pointer to the requested node for local nodes.

```
1  Node* get_node_by_gid(index gid):
2
3    if gid > max_gid:
4      throw UnknownNode()       // node does not exist
5
6    if gid == 0:
7      return local_nodes[0].node     // root node
8
9    if not ( local_min_gid ≤ gid ≤ local_max_gid ):
10     return 0                  // node not local
11
12   // estimate location in sparse node array
13   idx ← floor(1 + alpha * ( gid - local_min_gid ))
14
15   // search to the left
16   while 0 < idx and gid < local_nodes[idx].gid:
17     idx ← idx-1
18
19   // search to the right
20   while idx < local_nodes.size and local_nodes[idx].gid < gid:
21     idx ← idx+1
22
23   if idx < local_nodes.size and local_nodes[idx].gid == gid:
24     return local_nodes[idx].node    // local node found
25   else:
26     return 0                        // node not local
```

6.4 Appendix to Chap. 3: "Construction of the spiking network model"

6.4.1 Cortical areas in the model

Lobe	Abbreviation	Brain Region
Occipital	V1	Visual area 1
	V2	Visual area 2
	V3	Visual area 3
	VP	Ventral posterior
	V3A	Visual area V3A
	V4	Visual area 4
	VOT	Ventral occipitotemporal
	V4t	V4 transitional
	MT	Middle temporal
Temporal	FST	Floor of superior temporal
	PITd	Posterior inferotemporal (dorsal)
	PITv	Posterior inferotemporal (ventral)
	CITd	Central inferotemporal (dorsal)
	CITv	Central inferotemporal (ventral)
	AITd	Anterior inferotemporal (dorsal)
	AITv	Anterior inferotemporal (ventral)
	STPp	Superior temporal polysensory (posterior)
	STPa	Superior temporal polysensory (anterior)
	TF	Parahippocampal area TF
	TH	Parahippocampal area TH
Parietal	MSTd	Medial superior temporal (dorsal)
	MSTl	Medial superior temporal (lateral)
	PO	Parieto-occipital
	PIP	Posterior intraparietal
	LIP	Lateral intraparietal
	VIP	Ventral intraparietal
	MIP	Medial intraparietal
	MDP	Medial dorsal parietal
	DP	Dorsal prelunate
	7a	7a
Frontal	FEF	Frontal eye field
	46	Middle frontal area 46

Table 6.1: List of areas in the model. All vision-related areas of macaque cortex in the parcellation of Felleman & Van Essen (1991).

6.4.2 Inter-areal distances

Area	V1	V2	VP	V3	V3A	MT	V4t	V4	VOT	MSTd	PIP	PO	DP	MIP	MDP	VIP
V1	0.0	17.9	19.9	14.6	16.8	22.5	23.1	22.9	29.0	26.8	18.8	21.5	23.7	24.5	29.2	26.3
V2	17.9	0.0	16.1	17.8	18.2	20.0	20.5	21.2	24.5	24.4	19.8	23.8	24.6	26.0	30.8	25.8
VP	19.9	16.1	0.0	20.8	19.0	14.9	15.1	14.6	12.8	20.9	20.1	25.2	25.4	26.9	31.9	25.5
V3	14.6	17.8	20.8	0.0	8.1	15.9	17.0	18.5	26.9	19.4	10.6	14.6	15.1	16.9	22.0	18.1
V3A	16.8	18.2	19.0	8.1	0.0	12.4	13.4	15.6	23.4	15.4	9.2	15.0	9.4	16.3	21.2	13.8
MT	22.5	20.0	14.9	15.9	12.4	0.0	6.0	11.4	13.7	10.0	13.2	19.1	16.6	20.0	23.9	13.7
V4t	23.1	20.5	15.1	17.0	13.4	6.0	0.0	9.9	12.1	12.1	15.2	21.3	17.8	22.1	26.3	16.4
V4	22.9	21.2	14.6	18.5	15.6	11.4	9.9	0.0	13.1	17.8	18.6	24.6	20.4	25.6	30.5	21.4
VOT	29.0	24.5	12.8	26.9	23.4	13.7	12.1	13.1	0.0	19.7	24.6	30.0	28.5	31.0	36.0	26.6
MSTd	26.8	24.4	20.9	19.4	15.4	10.0	12.1	17.8	19.7	0.0	14.5	20.6	17.1	20.2	24.1	11.5
PIP	18.8	19.8	20.1	10.6	9.2	13.2	15.2	18.6	24.6	14.5	0.0	9.5	12.3	9.8	14.2	11.2
PO	21.5	23.8	25.2	14.6	15.0	19.1	21.3	24.6	30.0	20.6	9.5	0.0	18.9	6.9	10.0	16.0
DP	23.7	24.6	25.4	15.1	9.4	16.6	17.8	20.4	28.5	17.1	12.3	18.9	0.0	18.3	22.0	11.1
MIP	24.5	26.0	26.9	16.9	16.3	20.0	22.1	25.6	31.0	20.2	9.8	6.9	18.3	0.0	6.3	14.6
MDP	29.2	30.8	31.9	22.0	21.2	23.9	26.3	30.5	36.0	24.1	14.2	10.0	22.0	6.3	0.0	17.4
VIP	26.3	25.8	25.5	18.1	13.8	13.7	16.4	21.4	26.6	11.5	11.2	16.0	11.1	14.6	17.4	0.0
LIP	27.8	27.6	28.0	19.3	14.6	16.0	18.6	23.3	29.1	12.4	13.6	19.5	10.1	18.2	21.1	7.4
PITv	32.9	28.0	16.8	30.4	26.8	16.3	14.8	16.3	7.4	21.5	27.6	32.9	31.9	34.0	38.7	28.8
PITd	31.6	27.3	17.8	27.4	23.5	13.1	11.6	14.1	8.6	18.7	24.8	30.6	28.2	31.4	35.9	25.7
MSTl	28.4	24.4	17.4	22.6	18.9	8.2	9.9	15.7	13.2	9.1	18.0	24.0	22.7	24.2	28.3	16.9
CITv	38.8	33.4	22.4	36.0	32.7	21.6	21.1	22.5	12.9	25.7	32.4	37.6	37.7	38.8	43.1	33.1
CITd	37.7	32.5	22.0	34.3	31.0	19.5	19.2	20.9	12.2	23.2	30.5	35.9	35.7	37.0	41.0	30.7
FEF	57.1	53.9	48.3	50.0	45.8	37.9	40.3	45.8	42.1	36.4	42.9	47.9	46.2	45.4	46.7	36.8
TF	29.6	24.8	16.3	27.4	24.7	15.8	17.1	20.1	14.4	20.2	23.5	28.0	29.8	29.5	33.7	25.9
AITv	43.8	38.2	28.2	41.1	38.1	26.6	27.0	28.9	19.5	30.1	36.7	41.4	42.7	42.5	46.7	36.5
FST	33.7	28.9	20.3	28.8	25.6	14.2	15.7	19.2	12.8	16.5	24.6	29.8	29.6	30.6	34.2	23.8
7a	28.2	27.6	27.4	19.9	14.5	15.5	18.0	22.4	27.8	11.0	14.5	20.9	11.5	20.0	23.2	9.4
STPp	38.0	34.3	27.7	31.7	28.1	18.1	20.0	25.5	22.5	16.0	27.2	32.9	30.8	33.0	36.6	24.3
STPa	44.3	39.2	30.2	40.2	37.1	25.5	27.0	30.0	21.8	27.3	35.3	40.4	40.9	41.0	44.5	33.6
46	62.9	59.5	54.1	55.9	51.8	43.9	46.4	51.5	47.7	42.4	49.0	53.3	52.0	50.8	52.0	42.6
AITd	46.3	40.9	31.2	43.3	40.4	28.4	29.0	30.3	21.6	31.6	39.0	43.9	44.7	44.7	48.9	38.1
TH	30.8	26.3	19.9	27.5	25.1	17.1	18.9	22.6	18.1	20.6	22.4	26.2	29.4	28.2	31.4	24.9

Area	LIP	PITv	PITd	MSTl	CITv	CITd	FEF	TF	AITv	FST	7a	STPp	STPa	46	AITd	TH
V1	27.8	32.9	31.6	28.4	38.8	37.7	57.1	29.6	43.8	33.7	28.2	38.0	44.3	62.9	46.3	30.8
V2	27.6	28.0	27.3	24.4	33.4	32.5	53.9	24.8	38.2	28.9	27.6	34.3	39.2	59.5	40.9	26.3
VP	28.0	16.8	17.8	17.4	22.4	22.0	48.3	16.3	28.2	20.3	27.4	27.7	30.2	54.1	31.2	19.9
V3	19.3	30.4	27.4	22.6	36.0	34.3	50.0	27.4	41.1	28.8	19.9	31.7	40.2	55.9	43.3	27.5
V3A	14.6	26.8	23.5	18.9	32.7	31.0	45.8	24.7	38.1	25.6	14.5	28.1	37.1	51.8	40.4	25.1
MT	16.0	16.3	13.1	8.2	21.6	19.5	37.9	15.8	26.6	14.2	15.5	18.1	25.5	43.9	28.4	17.1
V4t	18.6	14.8	11.6	9.9	21.1	19.2	40.3	17.1	27.0	15.7	18.0	20.0	27.0	46.4	29.0	18.9
V4	23.3	16.3	14.1	15.7	22.5	20.9	45.8	20.1	28.9	19.2	22.4	25.5	30.0	51.5	30.3	22.6
VOT	29.1	7.4	8.6	13.2	12.9	12.2	42.1	14.4	19.5	12.8	27.8	22.5	21.8	47.7	21.6	18.1
MSTd	12.4	21.5	18.7	9.1	25.7	23.2	36.4	20.2	30.1	16.5	11.0	16.0	27.3	42.4	31.6	20.6
PIP	13.6	27.6	24.8	18.0	32.4	30.5	42.9	23.5	36.7	24.6	14.5	27.2	35.3	49.0	39.0	22.4
PO	19.5	32.9	30.6	24.0	37.6	35.9	47.9	28.0	41.4	29.8	20.9	32.9	40.4	53.3	43.9	26.2
DP	10.1	31.9	28.2	22.7	37.7	35.7	46.2	29.8	42.7	29.6	11.5	30.8	40.9	52.0	44.7	29.4
MIP	18.2	34.0	31.4	24.2	38.8	37.0	45.4	29.5	42.5	30.6	20.0	33.0	41.0	50.8	44.7	28.2
MDP	21.1	38.7	35.9	28.3	43.1	41.0	46.7	33.7	46.7	34.2	23.2	36.6	44.5	52.0	48.9	31.4
VIP	7.4	28.8	25.7	16.9	33.1	30.7	36.8	25.9	36.5	23.8	9.4	24.3	33.6	42.6	38.1	24.9
LIP	0.0	31.2	28.0	18.7	35.5	33.1	39.0	28.9	39.5	26.2	7.1	25.5	36.2	45.1	40.9	28.2
PITv	31.2	0.0	8.3	13.9	9.3	8.3	40.5	14.1	15.4	11.2	29.7	21.6	18.4	45.8	17.2	18.1
PITd	28.0	8.3	0.0	12.1	12.7	10.4	39.9	15.8	19.1	10.9	26.5	20.2	20.2	45.4	19.7	19.2
MSTl	18.7	13.9	12.1	0.0	17.7	15.1	32.4	14.3	22.2	8.8	17.2	11.9	19.8	38.4	23.9	15.8
CITv	35.5	9.3	12.7	17.7	0.0	6.4	38.7	14.6	9.3	11.8	33.9	20.9	13.8	43.2	10.9	18.3
CITd	33.1	8.3	10.4	15.1	6.4	0.0	36.5	13.9	10.3	9.0	31.4	18.5	12.4	41.2	10.7	17.2
FEF	39.0	40.5	39.9	32.4	38.7	36.5	0.0	39.9	36.5	30.5	39.4	33.3	29.3	11.2	35.2	39.8
TF	28.9	14.1	15.8	14.3	14.6	13.9	39.9	0.0	16.4	12.7	27.9	21.7	19.0	44.7	19.6	9.7
AITv	39.5	15.4	19.1	22.2	9.3	10.3	36.5	16.4	0.0	14.3	38.3	21.7	10.7	39.9	7.4	18.5
FST	26.2	11.2	10.9	8.8	11.8	9.0	30.5	12.7	14.3	0.0	24.7	12.4	12.2	36.0	15.5	14.6
7a	7.1	29.7	26.5	17.2	33.9	31.4	39.4	27.9	38.3	24.7	0.0	23.6	35.3	45.4	40.0	27.6
STPp	25.5	21.6	20.2	11.9	20.9	18.5	33.3	21.7	21.7	12.4	23.6	0.0	16.0	38.4	22.2	23.3
STPa	36.2	18.4	20.2	19.8	13.8	12.4	29.3	19.0	10.7	12.2	35.3	16.0	0.0	33.1	10.2	20.7
46	45.1	45.8	45.4	38.4	43.2	41.2	11.2	44.7	39.9	36.0	45.4	38.4	33.1	0.0	38.3	44.6
AITd	40.9	17.2	19.7	23.9	10.9	10.7	35.2	19.6	7.4	15.5	40.0	22.2	10.2	38.3	0.0	21.8
TH	28.2	18.1	19.2	15.8	18.3	17.2	39.8	9.7	18.5	14.6	27.6	23.3	20.7	44.6	21.8	0.0

Table 6.2: Distances between the areas of the model computed as the median of the distances between all vertex pairs of the two areas in their surface representation in F99 space, a standard macaque cortical surface included with Caret (Van Essen et al., 2001), where the vertex-to-vertex distance is the length of the shortest possible path without crossing the cortical surface (Bojak et al., 2011).

6.4.3 Neuron and synapse parameters

Synapse parameters		
Name	**Value**	**Description**
$J \pm \delta J$	Intra-areal connections: $87.8 \pm 8.8\,\mathrm{pA}$, cortico-cortical connections scaled as $J_{\mathrm{cc}} = \lambda J$, cortico-cortical connections onto inhibitory populations in addition scaled as $J^{\mathcal{I}}_{\mathrm{cc}} = \lambda_{\mathcal{I}} J^{\mathcal{E}}_{\mathrm{cc}}$ $\lambda \in [1, 2.5]$	excitatory synaptic strength
g	variable, $g \in [-12, -4]$	relative inhibitory synaptic strength
$d_{\mathrm{e}} \pm \delta d_{\mathrm{e}}$	$1.5 \pm 0.75\,\mathrm{ms}$	local excitatory transmission delay
$d_{\mathrm{i}} \pm \delta d_{\mathrm{i}}$	$0.75 \pm 0.375\,\mathrm{ms}$	local inhibitory transmission delay
$d \pm \delta d$	$d = s/v_{\mathrm{t}} \pm \frac{1}{2} s/v_{\mathrm{t}}$	inter-areal transmission delay, with s the distance between areas
v_{t}	$3.5\,\mathrm{m/s}$	transmission speed

Neuron model		
Name	**Value**	**Description**
τ_{m}	$10\,\mathrm{ms}$	membrane time constant
τ_{r}	$2\,\mathrm{ms}$	absolute refractory period
τ_{s}	$0.5\,\mathrm{ms}$	postsynaptic current time constant
C_{m}	$250\,\mathrm{pF}$	membrane capacity
V_r	$-65\,\mathrm{mV}$	reset potential
θ	$-50\,\mathrm{mV}$	fixed firing threshold
E_{L}	$-65\,\mathrm{mV}$	leak potential

Table 6.3: Parameter specification for single synapses and neurons.

6.4.4 Translation of Table 4 of Hilgetag et al. (2015)

Area in Hilgetag et al. (2016)	**FV91 area**
V1	V1
V2	V2
V3	V3
VP	VP
MT	MT
V3A	V3A
V4	V4
V4t	V4t
VOT	VOT
CIT	CITd, CITv
DP	DP
FEF	FEF
LIPd, LIPv	LIP
TEr*	AITd, AITv, CITd, CITv
MST	MSTd, MSTl
PIP	PIP
PIT	PITd, PITv
PO	PO
TF	TF
VIP	VIP
A46v	46
A7a	7a
AIT	AITd, AITv
FST	FST
STP	STPa, STPp
TH	TH
TEO*	PITd, PITv, VOT

Table 6.4: Scheme for translating architectural types, overall neuron densities and cortical thicknesses given in Table 4 of Hilgetag et al. (2016) to the modeled areas in the parcellation scheme of Felleman & Van Essen (1991). Entries marked with a star are used to translate the overall neuron density and cortical thickness which are not available in the finer of the two parcellations used by Hilgetag et al. (2016).

6.4.5 Relative laminar thicknesses from experimental literature

Area	1	2/3	4	5	6	Source
V1	0.08	0.25	0.37	0.14	0.16	O'Kusky & Colonnier (1982)
V1	0.09	0.29	0.39	0.11	0.12	Rakic et al. (1991)
V1	0.08	0.32	0.38	0.14	0.08	Felleman et al. (1997)
V1	0.05	0.31	0.36	0.14	0.14	Eggan & Lewis (2007)
V2	0.07	0.41	0.14	0.21	0.18	Markov et al. (2014a)
V2	0.1	0.42	0.19	0.13	0.16	Rakic et al. (1991)
V3	0.09	0.58	0.12	0.1	0.12	Markov et al. (2014a)
V3	0.2	0.29	0.27	nan	nan	Angelucci et al. (2002)
MT	0.11	0.54	0.13	0.11	0.11	Markov et al. (2014a)
MT	0.09	0.43	0.14	0.16	0.18	Preuss & Goldman-Rakic (1991)
V4	0.09	0.53	0.12	0.12	0.12	Rockland (1992)
MIP	0.09	0.41	0.08	0.08	0.34	Rozzi et al. (2006)
VIP	0.12	0.56	0.14	0.1	0.08	Preuss & Goldman-Rakic (1991)
LIP	0.09	0.36	0.09	0.08	0.39	Rozzi et al. (2006)
LIP	0.13	0.52	0.12	0.13	0.1	Preuss & Goldman-Rakic (1991)
FEF	0.1	0.42	0.16	0.17	0.16	Boussaoud et al. (1990)
TF	0.14	0.39	0.12	nan	nan	Preuss & Goldman-Rakic (1991)
FST	0.24	0.42	0.08	nan	nan	Lavenex et al. (2002)
46	0.1	0.45	0.1	0.15	0.2	Eggan & Lewis (2007)
46	0.13	0.43	0.09	nan	nan	Petrides & Pandya (1999)
TH	nan	nan	0.0	nan	nan	Suzuki & Amaral (2003)
TH	0.14	0.33	0.12	0.29	0.13	Preuss & Goldman-Rakic (1991)

Table 6.5: Relative laminar thicknesses determined from the anatomical studies given in the last column.

6.4.6 Laminar thicknesses

Area	1	2/3	4	5	6	Total
V1	0.09	0.37	0.46	0.17	0.16	1.24
V2	0.12	0.60	0.24	0.25	0.25	1.46
VP	0.18	0.63	0.32	0.21	0.25	1.59
V3	0.23	0.70	0.31	0.16	0.19	1.59
PIP	0.26	0.92	0.24	0.30	0.36	2.07
V3A	0.20	0.71	0.24	0.23	0.28	1.66
MT	0.20	0.95	0.26	0.26	0.29	1.96
V4t	0.22	0.80	0.29	0.26	0.31	1.88
V4	0.18	1.00	0.24	0.24	0.24	1.89
PO	0.26	0.92	0.24	0.30	0.36	2.07
VOT	0.23	0.81	0.28	0.27	0.32	1.90
DP	0.26	0.91	0.23	0.30	0.36	2.06
MIP	0.20	0.85	0.17	0.16	0.70	2.07
MDP	0.26	0.92	0.24	0.30	0.36	2.07
MSTd	0.26	0.92	0.24	0.30	0.36	2.07
VIP	0.25	1.17	0.28	0.21	0.16	2.07
LIP	0.25	1.00	0.24	0.24	0.57	2.30
PITv	0.23	0.81	0.28	0.27	0.32	1.90
PITd	0.23	0.81	0.28	0.27	0.32	1.90
AITv	0.34	1.20	0.23	0.39	0.47	2.63
MSTl	0.26	0.92	0.24	0.30	0.36	2.07
FST	0.51	0.90	0.18	0.30	0.36	2.25
CITv	0.29	1.02	0.19	0.33	0.40	2.23
CITd	0.29	1.02	0.19	0.33	0.40	2.23
7a	0.35	1.24	0.21	0.41	0.48	2.68
STPp	0.29	1.03	0.18	0.34	0.40	2.25
STPa	0.29	1.03	0.18	0.34	0.40	2.25
FEF	0.22	0.92	0.35	0.37	0.35	2.21
46	0.22	0.82	0.18	0.28	0.36	1.86
TF	0.23	0.66	0.21	0.24	0.28	1.62
TH	0.28	0.65	0.12	0.57	0.26	1.87
AITd	0.34	1.20	0.23	0.39	0.47	2.63

Table 6.6: Laminar thicknesses in mm for all 32 areas of the model. Values are rounded to two decimal places. These values are used to determine population sizes for the modeled layers 2/3, 4, 5 and 6 and to distribute synapses across layers 1 to 6 of target areas for cortico-cortical connections (cf. Results and Table 6.10).

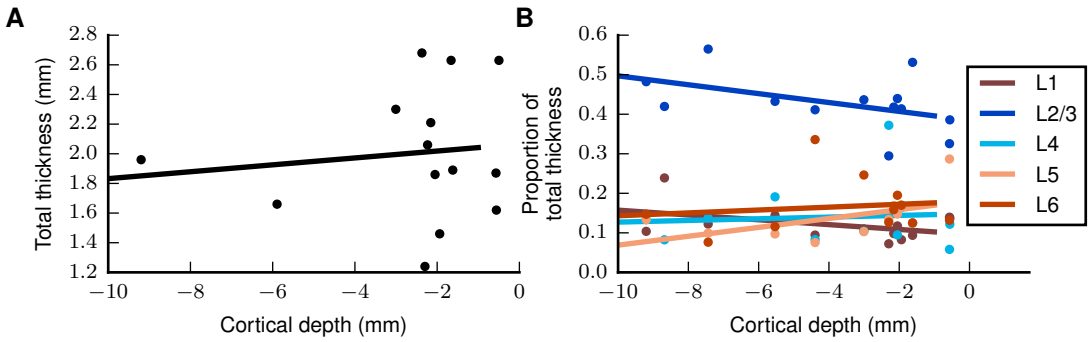

Figure 6.1: Related to Fig. 3.1. **Thickness versus cortical depth.** (A) Total thickness vs. cortical depth and linear least-squares fit showing no significant correlation ($r = 0.12$, $p = 0.68$). (B) Relative laminar thicknesses vs. cortical depth and linear least-squares fits also showing no significant correlation (L1: $r = -0.43, p = 0.14$, L2/3: $r = -0.46, p = 0.11$, L4: $r = 0.08$, $p = 0.79$; L5: $r = -0.53$, $p = 0.09$, L6: $r = 0.14, p = 0.69$). The thickness data is the same as in Fig. 3.1. Cortical depth data obtained from F99 surface statistics available through the Caret Software (Van Essen, 2012). Values for each area are averaged across cortical surface and both hemispheres. The data is obtained using the F99 Sulcal depth tool on http://cocomac.g-node.org and can be directly accessed via these two links: http://cocomac.g-node.org/cocomac2/services/f99_sulcal_depth.php?atlas=FV91&shape=Depth-Right&mode=avg&output=tsv&run=1 and http://cocomac.g-node.org/cocomac2/services/f99_sulcal_depth.php?atlas=FV91&shape=Depth-Left&mode=avg&output=tsv&run=1.

6.4.7 Area surfaces

Area	Surface area (mm^2)	Area	Surface area (mm^2)	Area	Surface area (mm^2)
V1	1484.63	V3	120.57	PO	75.37
V2	1193.40	CITv	114.67	VOT	70.11
V4	561.41	DP	113.83	LIP	56.04
STPp	245.48	PIP	106.15	MT	55.90
TF	197.40	PITv	100.34	FST	61.33
46	185.16	AITd	91.59	CITd	57.54
FEF	161.54	VIP	85.06	MIP	45.09
7a	157.34	V3A	96.96	TH	44.60
PITd	145.38	AITv	93.12	MSTl	29.19
VP	130.58	STPa	78.72	V4t	28.23
MSTd	120.57	MDP	77.49		

Table 6.7: Surface areas computed with Caret (Van Essen et al., 2001) on the basis of each area's representation on the F99 cortical surface (Van Essen, 2002). Areas are ordered from large to small.

6.4.8 Population sizes

Area	2/3E	2/3I	4E	4I	5E	5I	6E	6I	Total
V1	47386	13366	70387	17597	20740	4554	19839	4063	197935
V2	50521	14250	36685	9171	19079	4189	19248	3941	157087
VP	52973	14942	49292	12323	15929	3497	19130	3917	172007
V3	58475	16494	47428	11857	12056	2647	14529	2975	166465
PIP	44343	12507	22524	5631	14742	3237	17704	3625	124318
V3A	40887	11532	23789	5947	12671	2782	15218	3116	115946
MT	60606	17095	28202	7050	14176	3113	15837	3243	149324
V4t	48175	13588	34735	8684	14857	3262	17843	3654	144801
V4	64447	18178	33855	8464	13990	3072	14161	2900	159070
PO	44343	12507	22524	5631	14742	3237	17704	3625	124318
VOT	45313	12781	37611	9403	15828	3475	19008	3892	147315
DP	43934	12392	18896	4724	14179	3113	17028	3487	117755
MIP	41274	11642	15875	3969	7681	1686	34601	7086	123816
MDP	44343	12507	22524	5631	14742	3237	17704	3625	124318
MSTd	44343	12507	22524	5631	14742	3237	17704	3625	124318
VIP	56683	15988	26275	6569	10099	2217	7864	1610	127310
LIP	51983	14662	20095	5024	11630	2554	28115	5757	139824
PITv	45313	12781	37611	9403	15828	3475	19008	3892	147315
PITd	45313	12781	37611	9403	15828	3475	19008	3892	147315
AITv	49224	13884	18066	4516	16982	3729	20395	4176	130977
MSTl	44343	12507	22524	5631	14742	3237	17704	3625	124318
FST	36337	10249	12503	3126	12624	2772	15160	3104	95879
CITv	41696	11761	15303	3826	14385	3158	17275	3537	110944
CITd	41696	11761	15303	3826	14385	3158	17275	3537	110944
7a	49481	13957	13279	3320	15817	3473	18996	3890	122216
STPp	41677	11755	13092	3273	14218	3122	17075	3496	107712
STPa	41677	11755	13092	3273	14218	3122	17075	3496	107712
FEF	44053	12425	23143	5786	16943	3720	16128	3302	125504
46	32581	9190	10645	2661	11850	2602	15841	3244	88617
TF	30774	8680	17143	4286	11082	2433	13310	2725	90436
TH	24712	6970			23353	5128	10861	2224	73251
AITd	49224	13884	18066	4516	16982	3729	20395	4176	130977

Table 6.8: Estimated population sizes across layers and areas underneath $1\,\mathrm{mm}^2$ of cortical surface in each area.

6.4.9 Processing of CoCoMac data

We use a new release of CoCoMac, in which mappings from brain regions in other nomenclatures were scrutinized to ensure a consistent transfer of connections into the FV91 name space. The CoCoMac database provides information on laminar patterns on the source side from retrograde tracing studies as well as on the target side from anterograde tracing studies. The data was extracted by using the following link, which specifies all search options:

http://cocomac.g-node.org/cocomac2/services/connectivity_matrix.php?dbdate=20141022&AP=AxonalProjections_FV91&constraint=&origins=&terminals=&square=1&merge=max&laminar=both&format=json&cite=1

Furthermore, we obtained the numbers of confirmative studies for each area-level connection with the following link:
http://cocomac.g-node.org/cocomac2/services/connectivity_matrix.php?dbdate=20141022&AP=AxonalProjections_FV91&constraint=&origins=&terminals=&square=1&merge=count&laminar=off&format=json&cite=1

To process these data, we applied the following steps:

- A connection is assumed to exist if there is at least one confirmative study reporting it.
- A connection from layer 2/3 is modeled if CoCoMac indicates a connection from either or both of layers 2 and 3.
- In the database, some layers carry an 'X' indicating a connection of unknown strength. We interpret these as '2' (corresponding to medium connection strength).
- We take connection strengths in CoCoMac to represent numbers of synapses in orders of magnitude, i.e., the relative number of synapses N^{ν}_{syn} in layer ν of area A with connection strength $s(\nu)$ is computed as $N^{\nu}_{\text{syn}} = 10^{s(v)} / \sum_{v' \in A} 10^{s(\nu')}$.

6.4.10 Mapping of injection sites to FV91 parcellation

Monkey	M132 area	FV91 area	Monkey	M132 area	FV91 area
M88RH	V1	V1	M101LH	V2	V2
M121LH	V1	V1	M101RH	V2	V2
M81LH	V1	V1	M103LH	V2	V2
M85LH	V1	V1	M123LH	V4	V4
M85RH	V1	V1	M121RH	V4	V4
BB289RH	STPr	STPa	M119LH	TEO	V4
BB289LH	STPi	STPp	BB135LH	7A	7a
M90RH	STPc	STPp	M89LH	DP	DP
M106LH	9/46d	FEF	BB272RH	8l	FEF
M133LH	MT	MSTd	M116LH	46d	46
M116RH	9/46v	46	BB272LH	8m	FEF
M128RH	TEPd	CITv	M108LH	PBr	STPp

Table 6.9: Injected areas of the data set of Markov et al. (2014a) in the M132 parcellation and corresponding areas in the FV91 scheme. Only the injections in vision-related cortex are shown.

6.4.11 Mapping of synapse to cell-body locations

Detailed calculation in section Sec. 3.4. The numbers are listed in Table 6.10.

		Synapse layer				
		1	2/3	4	5	6
Target population	2/3E	0.57				
	2/3I		0.16			
	4E	0.18	0.84	0.73		
	4I			0.16		
	5E	0.25		0.02	0.76	
	5I				0.1	
	6E	0.003		0.09	0.14	0.85
	6I					0.15

Table 6.10: Conditional probabilities $\mathcal{P}(i|s_{\mathrm{cc}} \in v)$ for the target neuron to belong to population i if a cortico-cortical synapse s_{cc} is located in layer v, computed with Eq. 3.11 applied to the data set of Binzegger et al. (2004). Empty cells signal zero probabilities.

6.4.12 External input

Area	2/3E	2/3I	4E	4I	5E	5I	6E	6I
V1	1246	1246	1246	1246	1401	1246	1765	1246
V2	1848	1848	1848	1848	2079	1848	2618	1848
VP	1756	1756	1756	1756	1976	1756	2488	1756
V3	1810	1810	1810	1810	2036	1810	2564	1810
V3A	2703	2703	2703	2703	3041	2703	3830	2703
MT	2510	2510	2510	2510	2824	2510	3556	2510
V4t	2293	2293	2293	2293	2580	2293	3249	2293
V4	2337	2337	2337	2337	2630	2337	3311	2337
VOT	2409	2409	2409	2409	2710	2409	3413	2409
MSTd	3181	3181	3181	3181	3578	3181	4506	3181
PIP	3327	3327	3327	3327	3743	3327	4713	3327
PO	3226	3226	3226	3226	3629	3226	4570	3226
DP	3328	3328	3328	3328	3745	3328	4716	3328
MIP	3474	3474	3474	3474	3908	3474	4921	3474
MDP	5186	5186	5186	5186	5835	5186	7348	5186
VIP	3378	3378	3378	3378	3800	3378	4786	3378
LIP	3311	3311	3311	3311	3725	3311	4691	3311
PITv	2441	2441	2441	2441	2746	2441	3458	2441
PITd	2471	2471	2471	2471	2780	2471	3501	2471
MSTl	3094	3094	3094	3094	3481	3094	4383	3094
CITv	3844	3844	3844	3844	4324	3844	5446	3844
CITd	3708	3708	3708	3708	4172	3708	5253	3708
FEF	3597	3597	3597	3597	4047	3597	5096	3597
TF	3805	3805	3805	3805	4280	3805	5390	3805
AITv	3786	3786	3786	3786	4259	3786	5364	3786
FST	4614	4614	4614	4614	5191	4614	6537	4614
7a	4361	4361	4361	4361	4906	4361	6179	4361
STPp	4246	4246	4246	4246	4777	4246	6015	4246
STPa	4032	4032	4032	4032	4536	4032	5713	4032
46	4309	4309	4309	4309	4848	4309	6105	4309
AITd	3784	3784	3784	3784	4257	3784	5361	3784
TH	6590	5491			7413	5491	7780	5491

Table 6.11: Numbers of extrinsic synapses per neuron for all areas of the model with $\kappa = 1.125$.

6.5 Appendix to Chap. 5: "Dynamical results of the spiking network model"

6.5.1 Network simulations

We performed simulations on the JUQUEEN supercomputer (Jülich Supercomputing Centre, 2015) with NEST (Gewaltig & Diesmann, 2007) version 2.8.0 with optimizations for the use on the supercomputer which will enter one of the next releases. All simulations use a time step of 0.1 ms and exact integration for the subthreshold dynamics of the LIF neuron model (reviewed in Plesser & Diesmann, 2009). Simulations were run for 10.5**s** and 100.5**s** biological time discarding the first 500 ms. Spike times were recorded from all neurons, except for the simulations shown in Fig. 4.3 and Fig. 4.9, where we recorded from 1000 neurons per population.

6.5.2 Analysis methods

We investigate the structural properties of the model with the map equation method (Rosvall et al., 2010). In this clustering algorithm, an agent performs random walks between graph nodes with probability proportional to the outdegree of the present node and a probability ($p = 0.15$) of jumping to a random network node. The algorithm detects clusters in the graph by minimizing the length of a binary description of the network using a Huffman code. To assess the quality of the clustering, we use a modularity measure which extends a measure for unweighted, directed networks (Leicht & Newman, 2008) to weighted networks, analogous to Newman 2004,

$$Q = \frac{1}{m} \sum_{A,B} \left(\mathcal{O}_{AB} - \frac{\sum_{B'} \mathcal{O}_{AB'} \cdot \sum_{A'} K_{A'B}}{m} \right) \delta_{\mathcal{C}_A, \mathcal{C}_B} \,,$$

where $\mathcal{O}_{AB}(K_{AB})$ is the matrix of relative outdegrees (indegrees), $m = \sum_{A,B} \mathcal{O}_{A,B}$ and $\delta_{\mathcal{C}_A, \mathcal{C}_B} = 1$ if areas A and B are in the same cluster and 0 otherwise. $Q = 0$ reflects equal connectivity within and between clusters, while $Q = 1$ corresponds to connectivity exclusively within clusters.

Instantaneous firing rates are determined as spike histograms with bin width 1 ms averaged over the entire population or area. In Fig. 5.2G, Fig. 6.3G, and to determine the temporal hierarchy, we convolve the histograms with Gaussian kernels with $\sigma = 2$ ms. Spike-train irregularity is quantified for each population by the revised local variation LvR (Shinomoto et al., 2009) averaged over a subsample of 2000 neurons. The cross-correlation coefficient is computed with bin width 1 ms on single-cell spike histograms of a subsample of 2000 neurons per population with at least one emitted spike per neuron.

The single-cell autocorrelation function is calculated on spike histograms with bin width 2.5 ms to suppress fast fluctuations on the order of the refractory time, normalized to the zero-lag peak, and averaged across a subsample of 2000 neurons. We then perform a linear least-squares fit $f(t) = A - t/\tau$ on the logarithmized autocorrelation for all times $t \in [2.5, 75]$ and define the inverse slope τ as the intrinsic time scale of the population. If the autocorrelation drops to a local minimum at the first time lag $t = 2.5$ ms, we set the time scale to the refractory period, $\tau = 2$ ms.

The temporal hierarchy is based on the cross-covariance function between area-averaged firing rates. We use a wavelet-smoothing algorithm (`signal.find_peaks_cwt` of python scipy library (Jones et al., 2001) with peak width $\Delta = 20$ ms) to detect extrema for $\tau \in [-100, 100]$ and take the location of the extremum with the largest absolute value as the time lag.

Functional connectivity (FC) is defined as the zero-time lag cross-correlation coefficient of the area-averaged synaptic inputs

$$I_A(t) = \frac{1}{N_A} \sum_{i \in A} N_i \left| I_i(t) \right| = \frac{1}{N_A} \sum_{i \in A} N_i \sum_j K_{ij} \left| J_{ij} \right| \left(\nu_j * PSC_j \right)(t) \,,$$

with the normalized post-synaptic current $PSC_j(t) = \exp[-t/\tau_\mathrm{s}]$, the population firing rate ν_j of source population j, indegree K_{ij}, and synaptic weight J_{ij} of the connection from j to target population i containing N_i neurons.

The clustering of the FC matrices was performed using the function `modularity_louvain_und_sign` of the Brain Connectivity Toolbox (BCT; http://www.brain-connectivity-toolbox.net) with the Q^* option, which weights positive weights more strongly than negative weights, as introduced by Rubinov & Sporns (2011).

6.5.3 Supplemental Figures

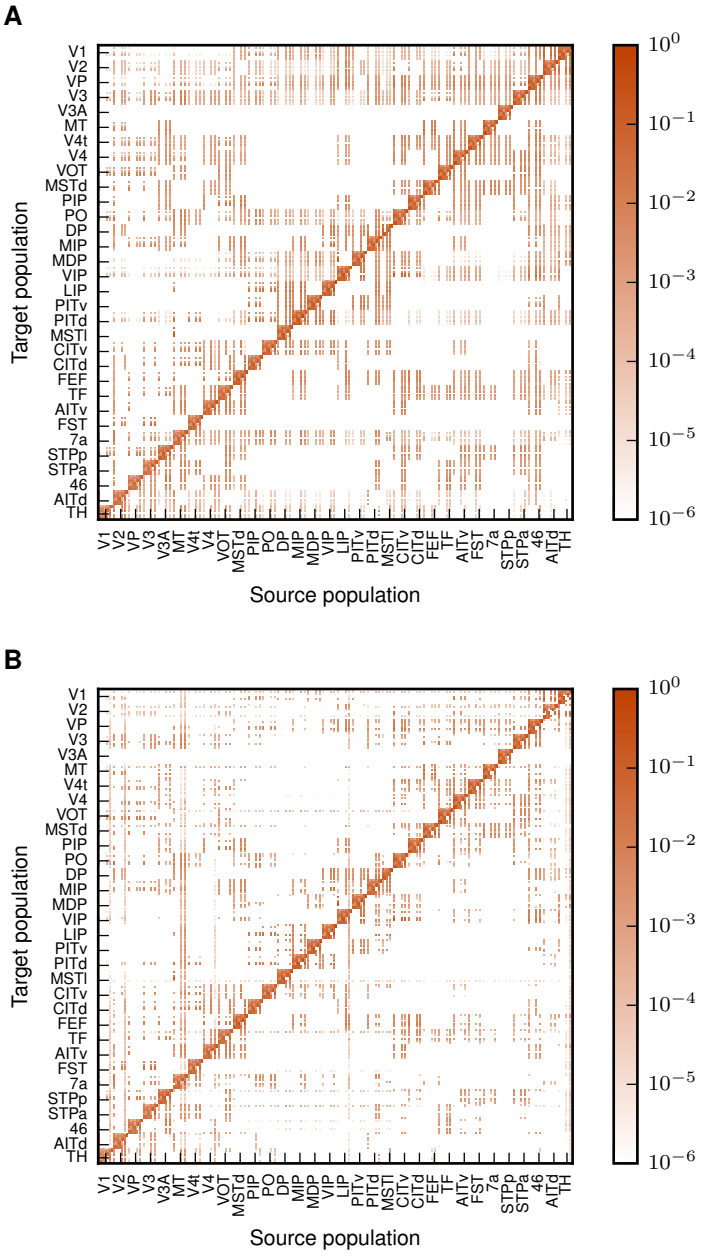

Figure 6.2: Connection probabilities of the model encoded in color before (A) and after (B) applying the theoretical method described in Schuecker et al. (2015b). Areas are ordered according to their architectural types, and populations inside the areas are ordered as [2/3E, 2/3I, 4E/, 4I, 5E, 5I, 6E, 6I].

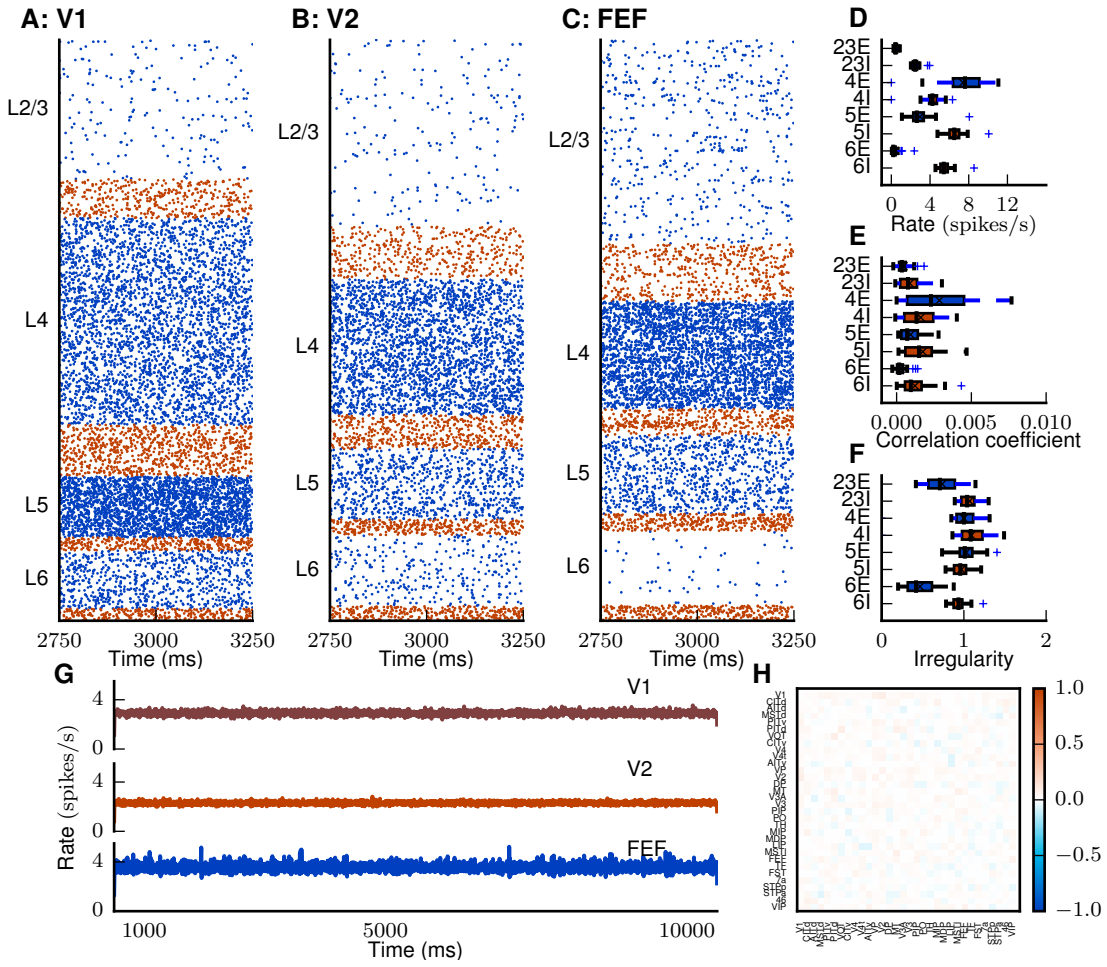

Figure 6.3: Related to Fig. 5.2. **Resting state of the model with cortico-cortical synaptic weights equal to intrinsic synaptic weights.** (A-C) Raster plot of spiking activity of $5\,\%$ of the neurons in area V1 (A), V2 (B), and PITv (C). (D-F) Statistics of the spiking activity across areas and populations shown as area-averaged box plots (Tukey, 1977). (D) Population-averaged firing rates. The firing rates differ across areas and layers, varying between 0.05 and 25 spikes/s. Inhibitory populations are more active than excitatory populations across layers and areas (with the exception of layer 5 in V1 and layer 4 in higher areas) despite the identical intrinsic properties of the two cell types. The excitatory populations of layer 2/3 and 6 exhibit lower firing rates than those of layers 4 and 5, similar to the microcircuit model (Potjans & Diesmann, 2014) (E) Average pairwise cross-correlation coefficients of spiking activity. (F) Irregularity measured by revised local variation LvR (Shinomoto et al., 2009) averaged across neurons. (G) Area-averaged firing rates. Parameters are $g = -11$, $\nu_{\text{ext}} = 10$ spikes/s, $\kappa = 1.125$, $\lambda = 1$, $\lambda_{\mathcal{I}} = 1$. (H) Functional connectivity (FC) in the model for measured by the zero-time lag cross-correlation coefficiens. Functional connectivity between areas is very low for most pairs of areas. Areas are ordered as in Fig. 5.6A.

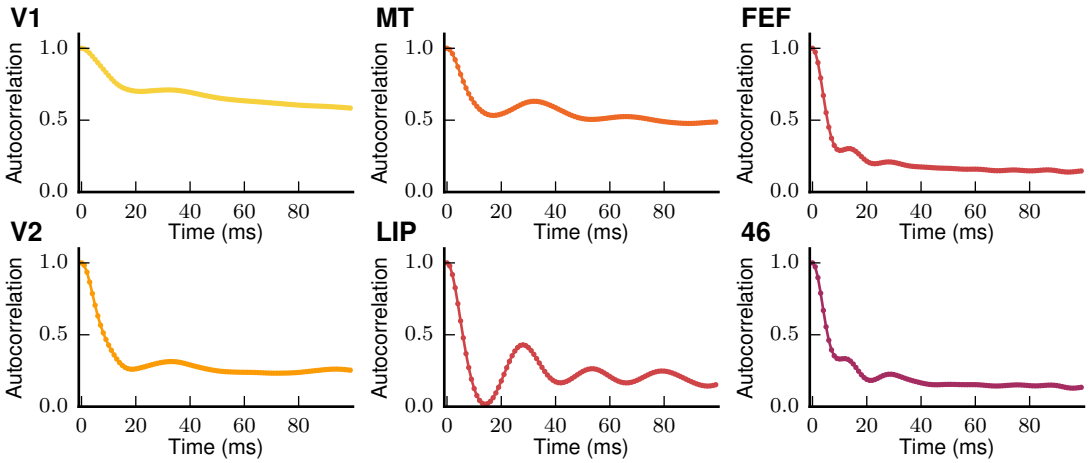

Figure 6.4: Related to Fig. 5.3. **Auto-correlation function of compound area rates** for the same areas as in Fig. 5.3. In contrast to the single-cell auto-correlations, the compound auto-correlations have time scales on the order of 20 ms already in early visual areas.

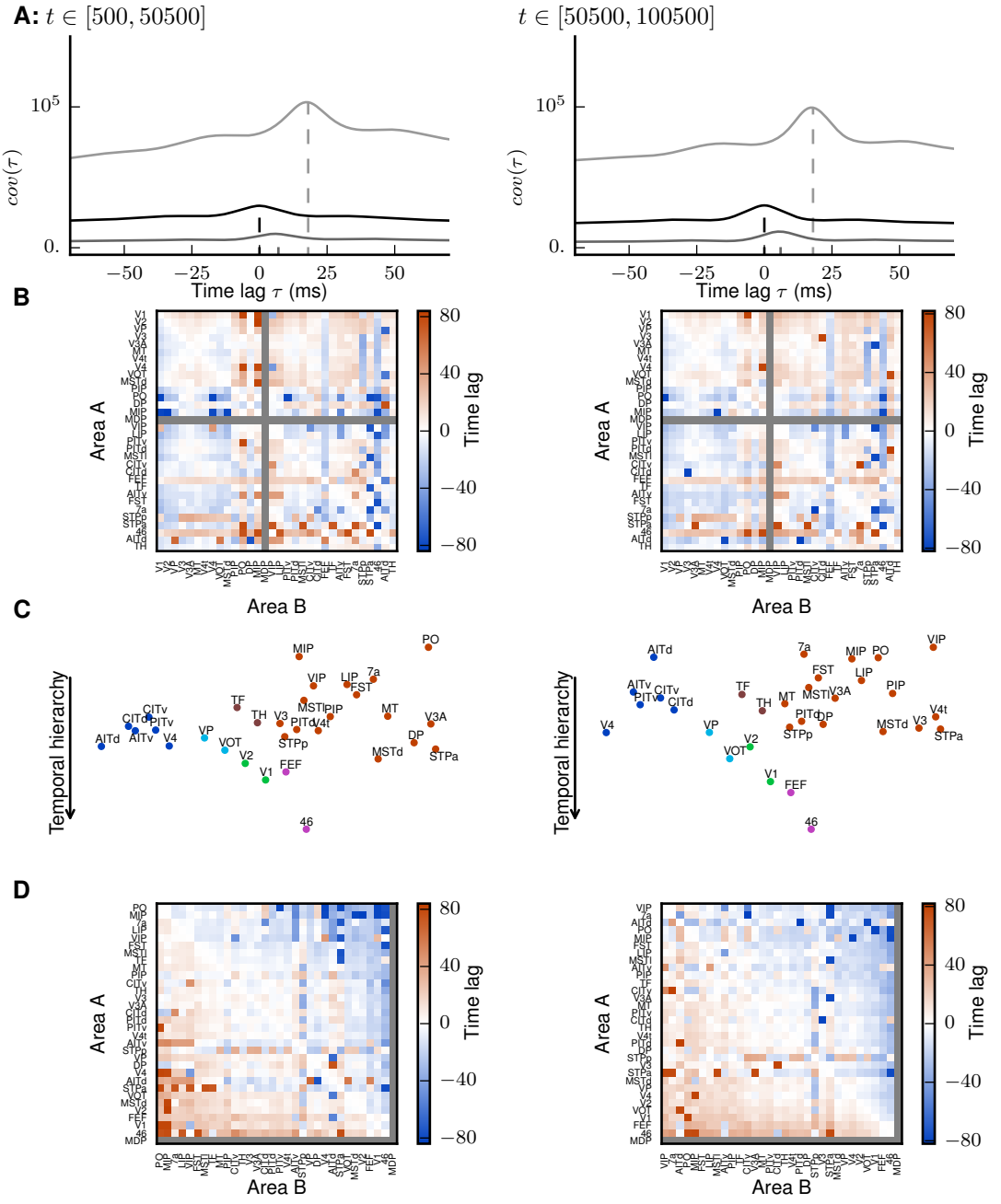

Figure 6.5: Related to Fig. 5.4. **Validation of the temporal hierarchy.** Left column: First half of the time series ($t \in [500\,\text{ms}, 50,500\,\text{ms}]$), right column: second half of the time series ($t \in [50,500\,\text{ms}, 100,500\,\text{ms}]$). (A) Covariance functions of the area-averaged firing rates of V1 with areas V2 (gray) and PITv (light gray), and auto-covariance function of V1 (black). Dashed lines mark peak positions, detected by a wavelet smoothing algorithm (see Sec. 6.5.2). (B) Matrix of peak positions of the correlation function for all pairs of areas. Area MDP was neglected because it has only outgoing but no incoming connections to other visual areas according to CoCoMac. (C) Temporal hierarchy. Colors indicate the cluster of each area found with the map equation algorithm (cf. Fig. 5.1). Areas are horizontally arranged to avoid visual overlap. (D) Peak position matrix with areas in hierarchical order.

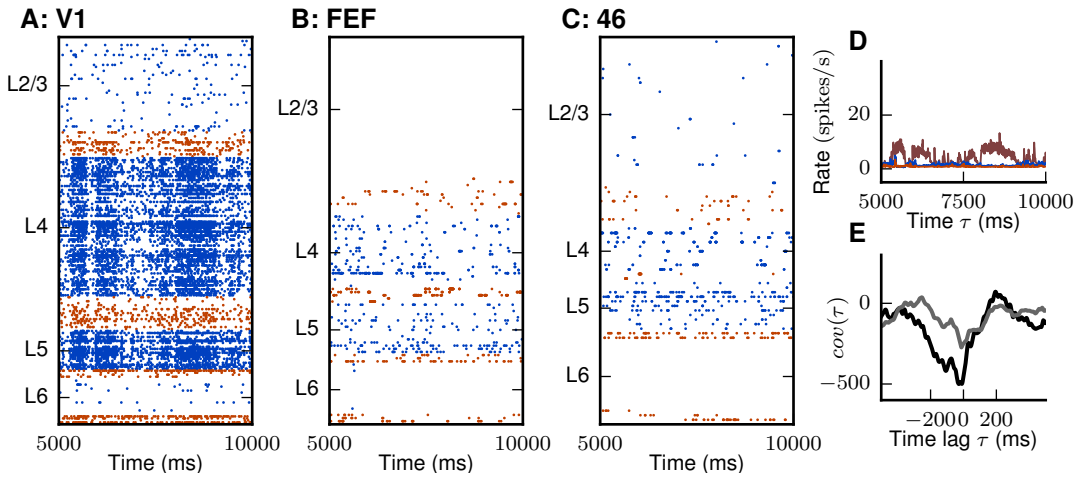

Figure 6.6: Related to Fig. 5.6. **Anticorrelation between V1 and frontal areas.** (A-C) Raster plot of spiking activity of $0.1\,\%$ of the neurons in area V1 (A), FEF (B), and 46 (C). (D) Area-averaged firing rates of V1 (brown), FEF (red) and 46 (blue). (E) Covariance functions of the area-averaged firing rates of V1 with areas 46 (black) and FEF (gray). Parameters are the same as for the simulation used in Fig. 5.6. The plot shows the anticorrelation between V1 and the two frontal areas visible in both the raster and the rate plot as well as in the negatively peaked cross-covariance function.

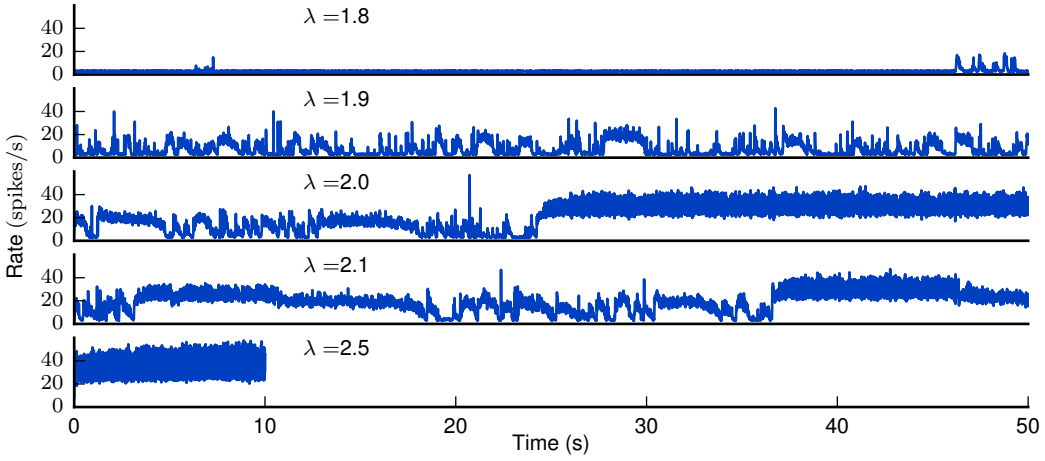

Figure 6.7: Related to Fig. 5.6. **Increasing cortico-cortical synaptic weights leads to switching to a high-activity state**. Area-averaged firing rates in V1 for four different settings of λ. The simulation for $\lambda = 2.5$ was run for 10s biological time only. From $\lambda = 2$ on, the network spontaneously enters a high-activity state. For $\lambda = 2.5$, the network is in this state from the outset.

Bibliography

Abramowitz, M. & Stegun, I. A. (1974). *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. New York: Dover Publications. (pp 28, 91, 144).

Aertsen, A. & Preißl, H. (1990). Dynamics of Activity and Connectivity in Physiological Neuronal Networks. In H. G. Schuster (Ed.), *Nonlinear Dynamics and Neuronal Networks*, Proceedings of the 63rd W. E. Heraeus Seminar Friedrichsdorf 1990 (pp. 281–301).: VCH. (p 129).

Ajima, Y., Sumimoto, S., & Shimizu, T. (2009). Tofu: A 6d mesh/torus interconnect for exascale computers. *Computer*, 42, 36–40. (p 47).

Alexandrescu, A. (2001). *Modern C++ Design: Generic Programming and Design Patterns Applied*. C++ In-Depth Series. Addison-Wesley Professional. (p 55).

Amit, D. J. & Brunel, N. (1997a). Dynamics of a recurrent network of spiking neurons before and following learning. *Network: Comput. Neural Systems*, 8, 373–404. (p 89).

Amit, D. J. & Brunel, N. (1997b). Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cereb. Cortex*, 7, 237–252. (p 92).

Ananthanarayanan, R., Esser, S. K., Simon, H. D., & Modha, D. S. (2009). The cat is out of the bag: Cortical simulations with 10^9 neurons and 10^{13} synapses. In *Supercomputing 09: Proceedings of the ACM/IEEE SC2009 Conference on High Performance Networking and Computing* Portland, OR. (p 65).

Ananthanarayanan, R. & Modha, D. S. (2007). Anatomy of a cortical simulator. In *Supercomputing 2007: Proceedings of the ACM/IEEE SC2007 Conference on High Performance Networking and Computing* New York, NY: Association for Computing Machinery. (p 124).

Anderson, J. C., Kennedy, H., & Martin, K. A. C. (2011). Pathways of attention: Synaptic relationships of frontal eye field to V4, lateral intraparietal cortex, and area 46 in macaque monkey. *J. Neurosci.*, 31(30), 10872–10881. (p 84).

Angelucci, A., Levitt, J. B., & Lund, J. S. (2002). Anatomical origins of the classical receptive field and modulatory surround field of single neurons in macaque visual cortical area V1. *Prog. Brain Res.*, 136, 373–388. (p 156).

Araya, R., Eisenthal, K. B., & Yuste, R. (2006a). Dendritic spines linearize the summation of excitatory potentials. *Proc. Natl. Acad. Sci. USA*, 103(49), 18799–18804. (p 19, 19).

Araya, R., Jiang, J., Eisenthal, K. B., & Yuste, R. (2006b). The spine neck filters membrane potentials. *Proceedings of the National Academy of Sciences*, 103(47), 17961–17966. (p 19).

Averbeck, B. B., Latham, P. E., & Pouget, A. (2006). Neural correlations, population coding and computation. *Nat. Rev. Neurosci.*, 7, 358–366. (p 17).

Axer, M., Grässel, D., Kleiner, M., Dammers, J., Dickscheid, T., Reckfort, J., Hütz, T., Eiben, B., Pietrzyk, U., Zilles, K., et al. (2011). High-resolution fiber tract reconstruction in the human brain by means of three-dimensional polarized light imaging. *Front. Neuroinformatics*, 5(34). (pp 15, 89).

Bakker, R., Thomas, W., & Diesmann, M. (2012). CoCoMac 2.0 and the future of tract-tracing databases. *Front. Neuroinformatics*, 6(30). (pp 4, 15, 78, 98, 124, 126).

Bakker, R., Tiesinga, P., & Kötter, R. (2015). The Scalable Brain Atlas: Instant web-based access to public brain atlases and related content. *Neuroinformatics*, 13, 353–366. (p 78).

Barbas, H. (1986). Pattern in the laminar origin of corticocortical connections. *Journal of Comparative Neurology*, 252(3), 415–422. (p 71).

Barbas, H. & Rempel-Clower, N. (1997). Cortical structure predicts the pattern of corticocortical connections. *Cereb. Cortex*, 7(7), 635–646. (pp 13, 71, 97).

Barnes, C. L. & Pandya, D. N. (1992). Efferent cortical connections of multimodal cortex of the superior temporal sulcus in the rhesus monkey. *J. Compar. Neurol.*, 318(2), 222–244. (pp 78, 81, 82, 98, 98).

Barone, P., Batardiere, A., Knoblauch, K., & Kennedy, H. (2000). Laminar distribution of neurons in extrastriate areas projecting to visual areas v1 and v4 correlates with the hierarchical rank and indicates the operation of a distance rule. *J. Neurosci.*, 20(9), 3263–3281. (p 16).

Bastos, A. M., Vezoli, J., Bosman, C. A., Schoffelen, J.-M., Oostenveld, R., Dowdall, J. R., De Weerd, P., Kennedy, H., & Fries, P. (2015a). Visual areas exert feedforward and

feedback influences through distinct frequency channels. *Neuron*, 85(2), 390–401. (pp 4, 18, 126).

Bastos, A. M., Vezoli, J., & Fries, P. (2015b). Communication through coherence with inter-areal delays. *Current opinion in neurobiology*, 31, 173–180. (p 18).

Belitski, A., Gretton, A., Magri, C., Murayama, Y., Montemurro, M. A., Logothetis, N. K., & Panzeri, S. (2008). Low-frequency local field potentials and spikes in primary visual cortex convey independent visual information. *J. Neurosci.*, 28(22), 5696–5709. (pp 18, 130).

Benda, J. & Herz, V. (2003). A universal model for spike-frequency adaptation. *Neural Comput.*, 15, 2523–2546. (p 127).

Bernacchia, A., Seo, H., Lee, D., & Wang, X.-J. (2011). A reservoir of time constants for memory traces in cortical neurons. *Nature neuroscience*, 14(3), 366–372. (p 17).

Berning, M., Boergens, K. M., & Helmstaedter, M. (2015). Segem: Efficient image analysis for high-resolution connectomics. *Neuron*, 87(6), 1193–1206. (p 13).

Beul, S. F., Barbas, H., & Hilgetag, C. C. (2015). A predictive structural model of the primate connectome. *arXiv preprint arXiv:1511.07222*. (p 80).

Beul, S. F. & Hilgetag, C. C. (2015). Towards a 'canonical' agranular cortical microcircuit. *Front. Neuroanatomy*, 8(165). (p 125).

Binzegger, T., Douglas, R. J., & Martin, K. A. C. (2004). A quantitative map of the circuit of cat primary visual cortex. *J. Neurosci.*, 39(24), 8441–8453. (pp 13, 25, 25, 28, 73, 73, 74, 81, 82, 82, 104, 125, 162).

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10), P10008. (pp 115, 118).

Board, O. A. R. (2008). *OpenMP Application Program Interface*. Specification, http://www.openmp.org/mp-documents/spec30.pdf. (p 47).

Bojak, I., Oostendorp, T. F., Reid, A. T., & Kötter, R. (2011). Towards a model-based integration of co-registered electroencephalography/functional magnetic resonance imaging data with realistic neural population meshes. *Phil. Trans. R. Soc. A*, 369(1952), 3785–3801. (pp 71, 153).

Bos, H., Diesmann, M., & Helias, M. (2015). Identifying anatomical origins of coexisting oscillations in the cortical microcircuit. *arXiv preprint arXiv:1510.00642*. (p 131).

Boussaoud, D., Ungerleider, L., & Desimone, R. (1990). Pathways for motion analysis: Cortical connections of the medial superior temporal and fundus of the superior temporal visual areas in the macaque. *J. Compar. Neurol.*, 296(3), 462–495. (p 156).

Braitenberg, V. & Schüz, A. (1998). *Cortex: Statistics and Geometry of Neuronal Connectivity*. Berlin: Springer-Verlag, 2nd edition. (p 13).

Branco, T. & Häusser, M. (2011). Synaptic integration gradients in single cortical pyramidal cell dendrites. *Neuron*, 69(5), 885 – 892. (p 19).

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci.*, 8(3), 183–208. (pp 39, 144).

Cabral, J., Hugues, E., Sporns, O., & Deco, G. (2011). Role of local network oscillations in resting-state functional connectivity. *NeuroImage*, 57(1), 130–139. (pp 3, 18, 29).

Cabral, J., Kringelbach, M. L., & Deco, G. (2014). Exploring the network dynamics underlying brain activity during rest. *Prog. Neurobiol.*, 114, 102–131. (p 89).

Calabrese, E., Badea, A., Coe, C. L., Lubach, G. R., Shi, Y., Styner, M. A., & Johnson, G. A. (2015a). A diffusion tensor MRI atlas of the postmortem rhesus macaque brain. *NeuroImage*, 117, 408–416. (p 125).

Calabrese, E., Badea, A., Cofer, G., Qi, Y., & Johnson, G. A. (2015b). A diffusion MRI tractography connectome of the mouse brain and comparison with neuronal tracer data. *Cereb. Cortex*, (pp. bhv121). (p 124).

Carnevale, T. & Hines, M. (2006). *The NEURON Book*. Cambridge: Cambridge University Press. (p 30).

Charvet, C. J., Cahalane, D. J., & Finlay, B. L. (2015). Systematic, cross-cortex variation in neuron numbers in rodents and primates. *Cerebral Cortex*, 25(1), 147–160. (p 124).

Chung, K. & Deisseroth, K. (2013). Clarity for mapping the nervous system. *Nature methods*, 10(6), 508–513. (p 13).

Cichocki, A., Zdunek, R., Phan, A. H., & Amari, S.-i. (2009). *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons. (p 91).

Colby, C., Gattass, R., Olson, C., & Gross, C. (1988). Topographical organization of cortical afferents to extrastriate visual area PO in the macaque: a dual tracer study. *J. Compar. Neurol.*, 269(3), 392–413. (pp 77, 130).

Cole, M. W., Bassett, D. S., Power, J. D., Braver, T. S., & Petersen, S. E. (2014). Intrinsic and task-evoked network architectures of the human brain. *Neuron*, 83(1), 238–251. (p 89).

Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jeffrey, D. J., & Knuth, D. E. (1996). On the lambert w function. *Advances in Computational Mathematics*, 5, 329–359. (p 45).

Cragg, B. (1967). The density of synapses and neurones in the motor and visual areas of the cerebral cortex. *J. Anat.*, 101(4), 639–654. (pp 74, 74, 124).

de Kock, C. P. J. & Sakmann, B. (2009). Spiking in primary somatosensory cortex during natural whisking in awake head-restrained rats is cell-type specific. *Proc. Natl. Acad. Sci. USA*, 106(38), 16446–16450. (pp 16, 98, 99).

De Pasquale, R. & Sherman, S. M. (2011). Synaptic properties of corticocortical connections between the primary and secondary visual cortical areas in the mouse. *J. Neurosci.*, 31(46), 16494–16506. (p 125).

Deco, G., Jirsa, V., McIntosh, A. R., Sporns, O., & Kötter, R. (2009). Key role of coupling, delay, and noise in resting brain fluctuations. *Proc. Natl. Acad. Sci. USA*, 106(25), 10302–10307. (pp 3, 29).

Deco, G. & Jirsa, V. K. (2012). Ongoing cortical activity at rest: Criticality, multistability, and ghost attractors. *J. Neurosci.*, 32(10), 3366–3375. (pp 3, 18, 29, 89).

Deco, G., Jirsa, V. K., & McIntosh, A. R. (2011). Emerging concepts for the dynamical organization of resting-state activity in the brain. *Nat. Rev. Neurosci.*, 12, 43–56. (p 18).

Dentico, D., Cheung, B. L., Chang, J.-Y., Guokas, J., Boly, M., Tononi, G., & Van Veen, B. (2014). Reversal of cortical information flow during visual imagery as compared to visual perception. *NeuroImage*, 100, 237–243. (p 129).

Destexhe, A., Rudolph, M., & Pare, D. (2003). The high-conductance state of neocortical neurons in vivo. *Nat. Rev. Neurosci.*, 4, 739–751. (p 26).

Diesmann, M. (2012). Brain-scale neuronal network simulations on k. In *Proceedings of the 4th Biosupercomputing Sympoisum* (pp. 83–85). Tokyo International Forum (Hall D7): Next-Generation Integrated Simulation of Living Matter (ISLiM) program of MEXT. (pp 39, 65).

Diesmann, M. (2013). The road to brain-scale simulations on K. *BioSupercomputing Newsletter*, 8, 8. (p 39).

Diesmann, M., Gewaltig, M.-O., & Aertsen, A. (1999). Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761), 529–533. (p 129).

Distler, C., Boussaoud, D., Desimone, R., & Ungerleider, L. G. (1993). Cortical connections of inferior temporal area teo in macaque monkeys. *J. Compar. Neurol.*, 334(1), 125–150. (p 82).

Dombrowski, S., Hilgetag, C., & Barbas, H. (2001). Quantitative architecture distinguishes prefrontal cortical systems in the rhesus monkey. *Cereb. Cortex*, 11(10), 975–988. (pp 13, 14, 71).

Ecker, A. S., Berens, P., Keliris, G. A., Bethge, M., & Logothetis, N. K. (2010). Decorrelated neuronal firing in cortical microcircuits. *Science*, 327(5965), 584–587. (p 17).

Eggan, S. & Lewis, D. (2007). Immunocytochemical distribution of the cannabinoid CB1 receptor in the primate neocortex: A regional and laminar analysis. *Cereb. Cortex*, 17(1), 175–191. (p 156, 156).

Elston, G. N. (2000). Pyramidal cells of the frontal lobe: all the more spinous to think with. *J. Neurosci.*, 20(18), RC95:1–4. (pp 124, 125).

Elston, G. N., Benavides-Piccione, R., Elston, A., Manger, P. R., & DeFelipe, J. (2011). Pyramidal cells in prefrontal cortex of primates: marked differences in neuronal structure among species. *Frontiers in Neuroanatomy*, 5(2). (pp 124, 125).

Elston, G. N. & Rosa, M. G. (2000). Pyramidal cells, patches, and cortical columns: a comparative study of infragranular neurons in TEO, TE, and the superior temporal polysensory area of the macaque monkey. *J. Neurosci.*, 20(24), RC117:1–5. (pp 124, 125).

Eppler, J. M. (2006). A multithreaded and distributed system for the simulation of large biological neural networks. Master's thesis, Albert Ludwig University Freiburg. (p 60).

Ercsey-Ravasz, M., Markov, N. T., Lamy, C., Essen, D. C. V., Knoblauch, K., Toroczkai, Z., & Kennedy, H. (2013). A predictive network model of cerebral cortical connectivity based on a distance rule. *Neuron*, 80(1), 184–197. (pp 4, 78).

Erdős, P. & Rényi, A. (1959). On random graphs I. *Publ. Math. Debrecen*, 6, 290–297. (p 22).

Felleman, D., Burkhalter, A., & Van Essen, D. (1997). Cortical connections of areas V3 and VP of macaque monkey extrastriate visual cortex. *J. Compar. Neurol.*, 379(1), 21–47. (p 156).

Felleman, D. J. & Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex*, 1, 1–47. (pp 10, 14, 14, 15, 16, 71, 73, 78, 79, 80, 81, 82, 82, 82, 82, 85, 98, 98, 98, 98, 101, 105, 126, 152, 155).

Fourcaud, N. & Brunel, N. (2002). Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural Comput.*, 14, 2057–2110. (pp 26, 28, 91, 91, 107, 144).

Fox, M., Corbetta, M., Snyder, A., Vincent, J., & Raichle, M. (2006). Spontaneous neuronal activity distinguishes human dorsal and ventral attention systems. *Proc. Nat. Acad. Sci. USA*, 103(26), 10046–10051. (p 18).

Fox, M. D. & Raichle, M. E. (2007). Spontaneous fluctuations in brain activity observed with functional magnetic resonance imaging. *Nature Reviews Neuroscience*, 8, 700–711. (pp 3, 18).

Fox, M. D., Snyder, A. Z., Vincent, J. L., Corbetta, M., Van Essen, D. C., & Raichle, M. E. (2005). The human brain is intrinsically organized into dynamic, anticorrelated functional networks. *Proc. Natl. Acad. Sci. USA*, 102(27), 9673–9678. (pp 3, 18).

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Theoretical Computer Science*, 9(10), 474–480. (p 18).

Fujisawa, S., Amarasingham, A., Harrison, M. T., & Buzsáki, G. (2008). Behavior-dependent short-term assembly dynamics in the medial prefrontal cortex. *Nat. Neurosci.*, 11(7), 823–833. (p 128).

Gabi, M., Collins, C. E., Wong, P., Torres, L. B., Kaas, J. H., & Herculano-Houzel, S. (2010). Cellular scaling rules for the brains of an extended number of primate species. *Brain, behavior and evolution*, 76(1), 32–44. (pp 4, 9).

Galletti, C., Kutz, D. F., Gamberini, M., Breveglieri, R., & Fattori, P. (2003). Role of the medial parieto-occipital cortex in the control of reaching and grasping movements. *Exp. Brain Res.*, 153(2), 158–170. (p 73).

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Professional Computing Series. Addison-Wesely. (p 55).

Gattass, R., Sousa, A., Mishkin, M., & Ungerleider, L. (1997). Cortical projections of area V2 in the macaque. *Cereb. Cortex*, 7(2), 110–129. (pp 77, 130).

Gewaltig, M.-O. & Diesmann, M. (2007). NEST (NEural Simulation Tool). *Scholarpedia*, 2(4), 1430. (pp 30, 37, 39, 85, 97, 164).

Girard, P., Hupé, J. M., & Bullier, J. (2001). Feedforward and feedback connections between areas v1 and v2 of the monkey have similar rapid conduction velocities. *J. Neurophysiol.*, 85(3), 1328–1331. (p 71).

Goulas, A., Bastiani, M., Bezgin, G., Uylings, H. B., Roebroeck, A., & Stiers, P. (2014). Comparative analysis of the macroscale structural connectivity in the macaque and human brain. *PLoS Comput. Biol.*, 10(3), e1003529. (p 89).

Haeusler, S., Schuch, K., & Maass, W. (2009). Motif distribution, dynamical properties, and computational performance of two data-based cortical microcircuit templates. *J. Physiol. (Paris)*, 103(1–2), 73–87. (pp 3, 3, 28).

Hagmann, P., Cammoun, L., Gigandet, X., Meuli, R., Honey, C. J., Wedeen, V. J., & Sporns, O. (2008). Mapping the structural core of human cerebral cortex. *PLoS Biol*, 6(7), e159. (p 29).

Hagmann, P., Kurant, M., Gigandet, X., Thiran, P., Wedeen, V. J., Meuli, R., & Thiran, J.-P. (2007). Mapping human whole-brain structural networks with diffusion mri. *PLOS ONE*, 2(7), e597. (p 15).

Hardy, G. & Wright, E. (1975). *An introduction to the theory of numbers*. Oxford at the Clarendon Press. (p 42).

Harrison, K. H., Hof, P. R., & Wang, S.-H. (2002). Scaling laws in the mammalian neocortex: Does form provide clues to function? *J. Neurocytol.*, 31, 289–298. (p 74).

Hawrylycz, M. J., Lein, E. S., Guillozet-Bongaarts, A. L., Shen, E. H., Ng, L., Miller, J. A., van de Lagemaat, L. N., Smith, K. A., Ebbert, A., Riley, Z. L., et al. (2012). An anatomically comprehensive atlas of the adult human brain transcriptome. *Nature*, 489(7416), 391–399. (p 128).

Helias, M., Kunkel, S., Masumoto, G., Igarashi, J., Eppler, J. M., Ishii, S., Fukai, T., Morrison, A., & Diesmann, M. (2012). Supercomputers ready for use as discovery machines for neuroscience. *Front. Neuroinform.*, 6, 26. (pp 4, 31, 37, 38, 39, 40, 42, 43, 48).

Helias, M., Tetzlaff, T., & Diesmann, M. (2013). Echoes in correlated neural systems. *New J. Phys.*, 15, 023002. (p 23).

Helias, M., Tetzlaff, T., & Diesmann, M. (2014). The correlation structure of local cortical networks intrinsically results from recurrent dynamics. *PLoS Comput. Biol.*, 10(1), e1003428. (p 17).

Helmstaedter, M. (2013). Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature methods*, 10(6), 501–507. (p 13).

Hennequin, G., Vogels, T., & Gerstner, W. (2012). Non-normal amplification in random balanced neuronal networks. *Phys. Rev. E*, 86, 011909. (p 131).

Hennequin, G., Vogels, T., & Gerstner, W. (2014). Optimal control of transient dynamics in balanced networks supports generation of complex movements. *Neuron*, 82, 1394–1406. (p 129).

Hilgetag, C. C. & Barbas, H. (2006). Role of mechanical factors in the morphology of the primate cerebral cortex. *PLoS Comput. Biol.*, 2(3), 146–159. (p 125).

Hilgetag, C. C., Medalla, M., Beul, S., & Barbas, H. (2016). The primate connectome in context: principles of connections of the cortical visual system. *NeuroImage*, (pp. (in press)). (pp 71, 71, 71, 73, 80, 81, 97, 155, 155, 155).

Hill, S. & Tononi, G. (2005). Modeling sleep and wakefulness in the thalamocortical system. *J. Neurophysiol.*, 93(3), 1671–1698. (pp 3, 29).

Hines, M. L., Markram, H., & Schürmann, F. (2008). Fully implicit parallel simulation of single neurons. *J. Comput. Neurosci.*, 25(3), 439–448. (p 124).

Hodgkin, A. L. & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. (Lond)*, 117, 500–544. (p 19).

Izhikevich, E. M. & Edelman, G. M. (2008). Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. USA*, 105(9), 3593–3598. (p 29).

Jadi, M., Polsky, A., Schiller, J., & Mel, B. W. (2012). Location-dependent effects of inhibition on local spiking in pyramidal neuron dendrites. *PLoS Comput Biol*, 8(6), e1002550. (p 19).

Jahnke, S., Memmesheimer, R.-M., & Timme, M. (2014). Hub-activated signal transmission in complex networks. *Phys. Rev. E*, 89, 030701. (p 129).

Jeffreys, H. & Jeffreys, B. (1956). *Methods of Mathematical Physics*. Cambridge University Press. (p 42).

Johnson, P. B., Ferraina, S., Bianchi, L., & Caminiti, R. (1996). Cortical networks for visual reaching: physiological and anatomical organization of frontal and parietal lobe arm regions. *Cereb. Cortex*, 6(2), 102–119. (p 73).

Johnson, R. R. & Burkhalter, A. (1996). Microcircuitry of forward and feedback connections within rat visual cortex. *J. Compar. Neurol.*, 368, 383–398. (p 84).

Johnson, R. R. & Burkhalter, A. (1997). A polysynaptic feedback circuit in rat visual cortex. *J. Neurosci.*, 17(18), 7129–7140. (p 84).

Jones, E., Coulter, J., & Hendry, S. (1978). Intracortical connectivity of architectonic fields in the somatic sensory, motor and parietal cortex of monkeys. *J. Compar. Neurol.*, 181(2), 291–347. (pp 82, 98).

Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python. http://www.scipy.org/. (p 165).

Jülich Supercomputing Centre (2015). JUQUEEN: IBM Blue Gene/Q® supercomputer system at the Jülich Supercomputing Centre. *Journal of large-scale research facilities*, 1. (pp 97, 164).

Kasthuri, N., Hayworth, K. J., Berger, D. R., Schalek, R. L., Conchello, J. A., Knowles-Barley, S., Lee, D., Vázquez-Reina, A., Kaynig, V., Jones, T. R., et al. (2015). Saturated reconstruction of a volume of neocortex. *Cell*, 162(3), 648–661. (p 13).

Knuth, D. E. (1997). *The Art of Computer Programming: Fundamental Algorithms*, volume 1. Addison Wesley, 3rd edition. (p 45).

Kozloski, J. & Wagner, J. (2011). An ultrascalable solution to large-scale neural tissue simulation. *Front. Neuroinform.*, 5(15). (p 124).

Kriener, B., Tetzlaff, T., Aertsen, A., Diesmann, M., & Rotter, S. (2008). Correlations and population dynamics in cortical networks. *Neural Comput.*, 20, 2185–2226. (p 114).

Krumnack, A., Reid, A. T., Wanke, E., Bezgin, G., & Kötter, R. (2010). Criteria for optimizing cortical hierarchies with continuous ranges. *Front. Neuroinformatics*, 4(7). (p 126).

Kuhn, A., Aertsen, A., & Rotter, S. (2004). Neuronal integration of synaptic input in the fluctuation-driven regime. *J. Neurosci.*, 24(10), 2345–2356. (p 127).

Kunkel, S., Helias, M., Potjans, T. C., Eppler, J. M., Plesser, H. E., Diesmann, M., & Morrison, A. (2012a). Memory consumption of neuronal network simulators at the brain scale. In K. Binder, G. Münster, & M. Kremer (Eds.), *NIC Symposium 2012 Proceedings*, volume 45 of *NIC Series* (pp. 81–88). Forschungszentrum Jülich GmbH. (pp 40, 122).

Kunkel, S., Potjans, T. C., Eppler, J. M., Plesser, H. E., Morrison, A., & Diesmann, M. (2012b). Meeting the memory challenges of brain-scale simulation. *Front. Neuroinform.*, 5, 35. (pp 37, 38, 38, 39, 40, 40, 41, 44, 48, 48, 59, 59, 65, 65).

Kunkel, S., Schmidt, M., Eppler, J., Plesser, H., Igarashi, J., Masumoto, G., Fukai, T., Ishii, S., Morrison, A., Diesmann, M., & Helias, M. (2013). From laptops to supercomputers: a single highly scalable code base for spiking neuronal network simulations. *BMC Neuroscience*, 14(Suppl 1), P163. (p 39).

Kunkel, S., Schmidt, M., Eppler, J. M., Masumoto, G., Igarashi, J., Ishii, S., Fukai, T., Morrison, A., Diesmann, M., & Helias, M. (2014). Spiking network simulation code for petascale computers. *Front. Neuroinformatics*, 8, 78. (p 35).

Lamme, V. A., Super, H., Spekreijse, H., et al. (1998). Feedforward, horizontal, and feedback processing in the visual cortex. *Curr. Opin. Neurobiol.*, 8(4), 529–535. (p 4).

Lapicque, L. (1907). Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *J. Physiol. Pathol. Gen.*, 9, 620–635. (p 19).

Lavenex, P., Suzuki, W., & Amaral, D. (2002). Perirhinal and parahippocampal cortices of the macaque monkey: Projections to the neocortex. *J. Compar. Neurol.*, 447(4), 394–420. (p 156).

Le Bon-Jego, M. & Yuste, R. (2007). Persistently active, pacemaker-like neurons in neocortex. *Front. Neurosci.*, 1(1), 123–129. (p 127).

Lee, W.-C. A., Huang, H., Feng, G., Sanes, J. R., Brown, E. N., So, P. T., & Nedivi, E. (2005). Dynamic remodeling of dendritic arbors in gabaergic interneurons of adult visual cortex. *PLoS Biol*, 4(2), e29. (p 11).

Leicht, E. A. & Newman, M. E. J. (2008). Community structure in directed networks. *Phys. Rev. Lett.*, 100, 118703. (p 164).

Lesnoff, M. & Lancelot, R. (2012). Analysis of overdispersed data, R package version 13. (p 81).

Lőrincz, M. L., Gunner, D., Bao, Y., Connelly, W. M., Isaac, J. T., Hughes, S. W., & Crunelli, V. (2015). A distinct class of slow (0.2–2 Hz) intrinsically bursting layer 5 pyramidal neurons determines UP/DOWN state dynamics in the neocortex. *J. Neurosci.*, 35(14), 5442–5458. (p 127).

Lidow, M. S. & Song, Z.-M. (2001). Primates exposed to cocaine in utero display reduced density and number of cerebral cortical neurons. *Journal of Comparative Neurology*, 435(3), 263–275. (pp 4, 9).

Lindén, H., Tetzlaff, T., Potjans, T. C., Pettersen, K. H., Grün, S., Diesmann, M., & Einevoll, G. T. (2011). Modeling the spatial reach of the LFP. *Neuron*, 72(5), 859–872. (p 17).

Logothetis, N. K., Pauls, J., Augath, M., Trinath, T., & Oeltermann, A. (2001). Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412, 150–157. (p 115).

Magnus, J. R. & Neudecker, H. (1995). *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons. (p 92).

Manual, G. I. & Stalhnan, R. (2013). *GCC 4.8.2 Manual*. Free Software Foundation. (p 58).

Markov, N. T., Ercsey-Ravasz, M., Van Essen, D. C., Knoblauch, K., Toroczkai, Z., & Kennedy, H. (2013). Cortical high-density counterstream architectures. *Science*, 342(6158). (pp 16, 89).

Markov, N. T., Ercsey-Ravasz, M. M., Ribeiro Gomes, A. R., Lamy, C., Magrou, L., Vezoli, J., Misery, P., Falchier, A., Quilodran, R., Gariel, M. A., Sallet, J., Gamanut, R., Huissoud, C., Clavagnier, S., Giroud, P., Sappey-Marinier, D., Barone, P., Dehay, C., Toroczkai, Z., Knoblauch, K., Van Essen, D. C., & Kennedy, H. (2014a). A weighted and directed interareal connectivity matrix for macaque cerebral cortex. *Cereb. Cortex*, 24(1), 17–36. (pp 4, 15, 15, 15, 78, 78, 78, 79, 89, 98, 98, 105, 124, 126, 156, 156, 156, 162).

Markov, N. T., Misery, P., Falchier, A., Lamy, C., Vezoli, J., Quilodran, R., Gariel, M. A., Giroud, P., Ercsey-Ravasz, M., Pilaz, L. J., Huissoud, C., Barone, P., Dehay, C., Toroczkai, Z., Van Essen, D. C., Kennedy, H., & Knoblauch, K. (2011). Weight consistency specifies regularities of macaque cortical networks. *Cereb. Cortex*, 21(6), 1254–1272. (pp 77, 79).

Markov, N. T., Vezoli, J., Chameau, P., Falchier, A., Quilodran, R., Huissoud, C., Lamy, C., Misery, P., Giroud, P., Ullman, S., Barone, P., Dehay, C., Knoblauch, K., & Kennedy, H. (2014b). Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex. *J. Compar. Neurol.*, 522(1), 225–259. (pp 4, 15, 15, 16, 18, 77, 80, 80, 80, 80, 81, 81, 85, 89, 98, 124, 124, 125, 130).

Markram, H., Muller, E., Ramaswamy, S., Reimann, M. W., Abdellah, M., Sanchez, C. A., Ailamaki, A., Alonso-Nanclares, L., Antille, N., Arsever, S., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2), 456–492. (pp 10, 12, 29, 74).

Marre, O., Yger, P., Davison, A. P., & Frégnac, Y. (2009). Reliable recall of spontaneous activity patterns in cortical networks. *J. Neurosci.*, 29(46), 14596–14606. (p 129).

Massimini, M., Huber, R., Ferrarelli, F., Hill, S., & Tononi, G. (2004). The sleep slow oscillation as a traveling wave. *J. Neurosci.*, 24(31), 6862–6870. (p 129).

McCulloch, C. E., Searle, S. R., & Neuhaus, J. M. (2008). *Generalized, Linear, and Mixed Models*. Wiley-Interscience, 2 edition. (p 80).

Message Passing Interface Forum (1994). *MPI: A Message-Passing Interface Standard.* Technical Report UT-CS-94-230. (p 47).

Mitra, A., Snyder, A. Z., Hacker, C. D., & Raichle, M. E. (2014). Lag structure in resting-state fmri. *J. Neurophysiol.*, 111(11), 2374–2391. (p 129).

Morel, A. & Bullier, J. (1990). Anatomical segregation of two cortical visual pathways in the macaque monkey. *Visual neuroscience*, 4(06), 555–578. (pp 81, 82, 98).

Morrison, A., Aertsen, A., & Diesmann, M. (2007). Spike-timing dependent plasticity in balanced random networks. *Neural Comput.*, 19, 1437–1467. (p 54).

Morrison, A., Diesmann, M., & Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike-timing. *Biol. Cybern.*, 98, 459–478. (pp 38, 58, 127).

Morrison, A., Mehring, C., Geisel, T., Aertsen, A., & Diesmann, M. (2005). Advancing the boundaries of high connectivity network simulation with distributed computing. *Neural Comput.*, 17(8), 1776–1801. (pp 30, 30, 57).

Murray, J. D., Bernacchia, A., Freedman, D. J., Romo, R., Wallis, J. D., Cai, X., Padoa-Schioppa, C., Pasternak, T., Seo, H., Lee, D., et al. (2014). A hierarchy of intrinsic timescales across primate cortex. *Nat. Neurosci.*, 17(12), 1661–1663. (pp 17, 113, 114, 114, 117, 128).

Nassi, J. J. & Callaway, E. M. (2009). Parallel processing strategies of the primate visual system. *Nat. Rev. Neurosci.*, 10, 360–372. (p 16).

Neske, G. T., Patrick, S. L., & Connors, B. W. (2015). Contributions of diverse excitatory and inhibitory neurons to recurrent network activity in cerebral cortex. *J. Neurosci.*, 35(3), 1089–1105. (p 127).

Newman, M. E. J. (2004). Analysis of weighted networks. *Phys. Rev. E*, 70, 056131. (pp 115, 164).

Nir, Y., Staba, R. J., Andrillon, T., Vyazovskiy, V. V., Cirelli, C., Fried, I., & Tononi, G. (2011). Regional slow waves and spindles in human sleep. *Neuron*, 70(1), 153–169. (p 129).

Nordlie, E., Gewaltig, M.-O., & Plesser, H. E. (2009). Towards reproducible descriptions of neuronal network models. *PLoS Comput. Biol.*, 5(8), e1000456. (pp 25, 72).

O'Kusky, J. & Colonnier, M. (1982). A laminar analysis of the number of neurons, glia, and synapses in the visual cortex (area 17) of adult macaque monkeys. *J. Compar. Neurol.*, 210(3), 278–290. (pp 74, 74, 124, 156).

Ostojic, S., Brunel, N., & Hakim, V. (2009). How connectivity, background activity, and synaptic properties shape the cross-correlation between spike trains. *J. Neurosci.*, 29(33), 10234–10253. (p 89).

Papoulis, A. & Pillai, S. U. (2002). *Probability, Random Variables, and Stochastic Processes*. Boston: McGraw-Hill, 4th edition. (pp 22, 138).

Pascual-Leone, A. & Walsh, V. (2001). Fast backprojections from the motion to the primary visual area necessary for visual awareness. *Science*, 292, 510–512. (p 4).

Perkel, D. J., Bullier, J., & Kennedy, H. (1986). Topography of the afferent connectivity of area 17 in the macaque monkey: A double-labelling study. *J. Compar. Neurol.*, 253(3), 374–402. (pp 81, 98).

Pernice, V., Staude, B., Cardanobile, S., & Rotter, S. (2011). How structure determines correlations in neuronal networks. *PLoS Comput. Biol.*, 7(5), e1002059. (p 89).

Peters, A. & Feldman, M. L. (1976). The projection of the lateral geniculate nucleus to area 17 of the rat cerebral cortex. I. General description. *J. Neurocytol.*, 5, 63–84. (p 13).

Petrides, M. & Pandya, D. (1999). Dorsolateral prefrontal cortex: comparative cytoarchitectonic analysis in the human and the macaque brain and corticocortical connection patterns. *Eur. J. Neurosci.*, 11(3), 1011–1036. (p 156).

Plesser, H. E. & Diesmann, M. (2009). Simplicity and efficiency of integrate-and-fire neuron models. *Neural Comput.*, 21, 353–359. (pp 30, 97, 164).

Plesser, H. E., Eppler, J. M., Morrison, A., Diesmann, M., & Gewaltig, M.-O. (2007). Efficient parallel simulation of large-scale neuronal networks on clusters of multiprocessor computers. In A.-M. Kermarrec, L. Bougé, & T. Priol (Eds.), *Euro-Par 2007: Parallel Processing*, volume 4641 of *Lecture Notes in Computer Science* (pp. 672–681). Berlin: Springer-Verlag. (pp 31, 60).

Potjans, T. C. & Diesmann, M. (2013). Multi-population network models of the cortical microcircuit. In *Advances in Cognitive Neurodynamics (III)* (pp. 91–96). Springer. (p 25).

Potjans, T. C. & Diesmann, M. (2014). The cell-type specific cortical microcircuit: Relating structure and activity in a full-scale spiking network model. *Cereb. Cortex*, 24(3), 785–806. (pp 5, 16, 21, 25, 25, 27, 29, 71, 71, 72, 74, 75, 75, 75, 76, 84, 97, 100, 104, 105, 124, 126, 129, 167).

Preuss, T. M. & Goldman-Rakic, P. S. (1991). Myelo-and cytoarchitecture of the granular frontal cortex and surrounding regions in the strepsirhine primate galago and the anthropoid primate macaca. *J. Compar. Neurol.*, 310(4), 429–474. (p 156, 156, 156, 156, 156).

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. (p 81).

Rakic, P., Suñer, I., & Williams, R. (1991). A novel cytoarchitectonic area induced experimentally within the primate visual cortex. *Proc. Nat. Acad. Sci. USA*, 88(6), 2083–2087. (p 156, 156).

Rao, R. P. N. & Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive field effects. *Nat. Neurosci.*, 2(1), 79–87. (p 4).

Rasch, M. J., Schuch, K., Logothetis, N. K., & Maass, W. (2011). Statistical comparison of spike responses to natural stimuli in monkey area V1 with simulated responses of a detailed laminar network model for a patch of V1. *J. Neurophysiol.*, 105, 757–778. (p 3).

Reid, A. T., Krumnack, A., Wanke, E., & Kötter, R. (2009). Optimization of cortical hierarchies with continuous scales and ranges. *NeuroImage*, 47, 611–617. (pp 16, 115).

Rentzsch, J. (2005). Data alignment: Straighten up and fly right. IBM developerWorks. (p 58).

Riesenhuber, M. & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2(11), 1019–1025. (p 4).

RIKEN BSI (2013). Largest neuronal network simulation achieved using K computer. Press release. (pp 31, 39, 64, 67, 122).

Risken, H. (1996). *The Fokker-Planck Equation*. Springer Verlag Berlin Heidelberg. (p 139).

Rockland, K. (1992). Configuration, in serial reconstruction, of individual axons projecting from area V2 to V4 in the macaque monkey. *Cereb. Cortex*, 2(5), 353–374. (p 156).

Rockland, K. S. (2004). Feedback connections: Splitting the arrow. In J. H. Kaas & C. E. Collins (Eds.), *The Primate Visual System* (pp. 387–406). CRC Press. (p 124).

Rockland, K. S., Andresen, J., Cowie, R. J., & Robinson, D. L. (1999). Single axon analysis of pulvinocortical connections to several visual areas in the macaque. *Journal of Comparative Neurology*, 406(2), 221–250. (p 85).

Rockland, K. S. & Pandya, D. N. (1979). Laminar origins and terminations of cortical connections of the occipital lobe in the rhesus monkey. *Brain Res.*, 179, 3–20. (pp 78, 82, 98, 98).

Rosvall, M., Axelsson, D., & Bergstrom, C. T. (2010). The map equation. *The European Physical Journal Special Topics*, 178(1), 13–23. (pp 112, 164).

Rotter, S. & Diesmann, M. (1999). Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biol. Cybern.*, 81(5/6), 381–402. (p 30).

Rozzi, S., Calzavara, R., Belmalih, A., Borra, E., Gregoriou, G., Matelli, M., & Luppino, G. (2006). Cortical connections of the inferior parietal cortical convexity of the macaque monkey. *Cereb. Cortex*, 16(10), 1389–1417. (p 156, 156).

Rubinov, M. & Sporns, O. (2011). Weight-conserving characterization of complex functional brain networks. *Neuroimage*, 56(4), 2068–2079. (p 165).

Sakata, S. & Harris, K. D. (2009). Laminar structure of spontaneous and sensory-evoked population activity in auditory cortex. *Neuron*, 64, 404–418. (pp 16, 128).

Salin, P., Bullier, J., & Kennedy, H. (1989). Convergence and divergence in the afferent projections to cat area 17. *J. Compar. Neurol.*, 283(4), 486–512. (pp 77, 130).

Salin, P.-A. & Bullier, J. (1995). Corticocortical connections in the visual system: structure and function. *Physiol. Rev.*, 75(1), 107–154. (pp 82, 101).

Scannell, J. W., Grant, S., Payne, B. R., & Baddeley, R. (2000). On variability in the density of corticocortical and thalamocortical connections. *Phil. Trans. R. Soc. B*, 355(1393), 21–35. (p 105).

Schmidt, M., Bakker, R., Shen, K., Bezgin, G., Diesmann, M., & van Albada, S. J. (2015a). Full-density multi-scale account of structure and dynamics of macaque visual cortex. *arXiv preprint arXiv:1511.09364*. (pp 21, 69, 98, 98, 109).

Schmidt, M., Schuecker, J., Diesmann, M., & Helias, M. (2015b). Shaping phase space of neural networks via connectivity. In *Proceedings of the 11th Göttingen Meeting of the German Neuroscience Society* (pp. T26 7C). (pp 87, 90).

Schmidt, M., van Albada, S., Bakker, R., & Diesmann, M. (2013). Integrating multi-scale data for a network model of macaque visual cortex. *BMC Neuroscience*, 14(Suppl 1), P111. (pp 69, 109).

Schmidt, M., van Albada, S., Bakker, R., & Diesmann, M. (2014). A spiking multi-area network model of macaque visual cortex. In *2014 Neuroscience Meeting Planner. Washington, DC: Society for Neuroscience.* (pp. 186.22/TT43). (pp 69, 109).

Schmitt, O., Eipert, P., Kettlitz, R., Leßmann, F., & Wree, A. (2014). The connectome of the basal ganglia. *Brain Structure and Function*, (pp. 1–62). (p 126).

Schuecker, J., Diesmann, M., & Helias, M. (2015a). Modulated escape from a metastable state driven by colored noise. *Phys. Rev. E*, 92, 052119. (p 144).

Schuecker, J., Schmidt, M., van Albada, S., Diesmann, M., & Helias, M. (2015b). Fundamental activity constraints lead to specific interpretations of the connectome. *arXiv preprint arXiv:1509.03162.* (pp 5, 21, 87, 166).

Schüz, A. & Palm, G. (1989). Density of neurons and synapses in the cerebral cortex of the mouse. *J. Compar. Neurol.*, 286, 442–455. (p 74).

Seltzer, B. & Pandya, D. N. (1991). Post-rolandic cortical projections of the superior temporal sulcus in the rhesus monkey. *J. Compar. Neurol.*, 312(4), 625–640. (p 98).

Seltzer, B. & Pandya, D. N. (1994). Parietal, temporal, and occipita projections to cortex of the superior temporal sulcus in the rhesus monkey: A retrograde tracer study. *J. Compar. Neurol.*, 343(3), 445–463. (pp 81, 98).

Shen, K., Bezgin, G., Hutchison, R., Gati, J., Menon, R., Everling, S., & McIntosh, R. (2012). Information processing architecture of functionally defined clusters in the macaque cortex. *J. Neurosci.*, 32(48), 17465–17476. (pp 18, 130, 131).

Shen, K., Hutchison, R. M., Bezgin, G., Everling, S., & McIntosh, A. R. (2015). Network structure shapes spontaneous functional connectivity dynamics. *J. Neurosci.*, 35(14), 5579–5588. (pp 18, 89, 117, 130, 131).

Sheng, T. (1985). The distance between two random points in plane regions. *Adv. Appl. Prob.*, 17(4), 748–773. (p 75).

Sheroziya, M. & Timofeev, I. (2014). Global intracellular slow-wave dynamics of the thalamocortical system. *J. Neurosci.*, 34(26), 8875–8893. (p 129).

Shinomoto, S., Kim, H., Shimokawa, T., Matsuno, N., Funahashi, S., Shima, K., Fujita, I., Tamura, H., Doi, T., Kawano, K., Inaba, N., Fukushima, K., Kurkin, S., Kurata, K., Taira, M., Tsutsui, K.-I., Komatsu, H., Ogawa, T., Koida, K., Tanji, J., & Toyama, K. (2009). Relating neuronal firing patterns to functional differentiation of cerebral cortex. *PLoS Comput. Biol.*, 5(7), e1000433. (pp 113, 145, 164, 167).

Shoham, S., O'Connor, D. H., & Segev, R. (2006). How silent is the brain: is there a "dark matter" problem in neuroscience? *J. Comp. Phys.*, 192, 777–784. (p 17).

Siegert, A. J. (1951). On the first passage time probability problem. *Phys. Rev.*, 81(4), 617–623. (pp 28, 144).

Stein, R. B. (1967). Some models of neuronal variability. *Biophys. J.*, 7, 37–68. (p 19).

Stephan, K., Kamper, L., Bozkurt, A., Burns, G., Young, M., & Kötter, R. (2001). Advanced database methodology for the collation of connectivity data on the macaque brain (CoCoMac). *Phil. Trans. R. Soc. B*, 356, 1159–1186. (pp 4, 15, 29, 78, 98, 124, 126).

Strogatz, S. H. (1994). *Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering.* Reading, Massachusetts: Perseus Books. (pp 93, 101).

Stroustrup, B. (1997). *The C++ Programming Language.* New York: Addison-Wesely, 3 edition. (pp 57, 59).

Sussillo, D., Toyoizumi, T., & Maass, W. (2007). Self-tuning of neural circuits through short-term synaptic plasticity. *J. Neurophysiol.*, 97, 4079–4095. (p 129).

Suzuki, W. A. & Amaral, D. G. (1994a). Topographic organization of the reciprocal connections between the monkey entorhinal cortex and the perirhinal and parahippocampal cortices. *J. Neurosci.*, 14(3), 1856–1877. (p 78).

Suzuki, W. A. & Amaral, D. G. (2003). Perirhinal and parahippocampal cortices of the macaque monkey: cytoarchitectonic and chemoarchitectonic organization. *J. Compar. Neurol.*, 463(1), 67–91. (p 156).

Suzuki, W. L. & Amaral, D. G. (1994b). Perirhinal and parahippocampal cortices of the macaque monkey: cortical afferents. *J. Compar. Neurol.*, 350(4), 497–533. (pp 81, 82, 98).

Swadlow, H. A. (1988). Efferent neurons and suspected interneurons in binocular visual cortex of the awake rabbit: Receptive fields and binocular properties. *J. Neurophysiol.*, 59(4), 1162–1187. (pp 16, 16, 98, 99, 128).

Tetzlaff, T., Helias, M., Einevoll, G., & Diesmann, M. (2012). Decorrelation of neural-network activity by inhibitory feedback. *PLoS Comput. Biol.*, 8(8), e1002596. (p 17).

Thomas, C., Frank, Q. Y., Irfanoglu, M. O., Modi, P., Saleem, K. S., Leopold, D. A., & Pierpaoli, C. (2014). Anatomical accuracy of brain connections derived from diffusion mri tractography is inherently limited. *Proc. Natl. Acad. Sci. USA*, 111(46), 16574–16579. (pp 15, 124).

Thomson, A. M. & Lamy, C. (2007). Functional maps of neocortical local circuitry. *Front. Neurosci.*, 1(1), 19–42. (pp 13, 25, 25, 74).

Thomson, A. M., West, D. C., Wang, Y., & Bannister, A. (2002). Synaptic connections and small circuits involving excitatory and inhibitory neurons in layer 2-5 of adult rat and cat neocortex: Triple intracellular recordings and biocytin labelling in vitro. *Cereb Cortex*, 12, 936–953. (p 28).

Tomioka, R. & Rockland, K. S. (2007). Long-distance corticocortical GABAergic neurons in the adult monkey white and gray matter. *J. Compar. Neurol.*, 505(5), 526–538. (pp 82, 101).

Traub, R. D., Contreras, D., Cunningham, M. O., Murray, H., LeBeau, F. E. N., Roopun, A., Bibbig, A., Wilent, W. B., Higley, M. J., & Whittington, M. A. (2005). Single-column thalamocortical network model exhibiting gamma oscillations, sleep spindles, and epileptogenic bursts. *J. Neurophysiol.*, 93(4), 2194–2232. (pp 3, 28).

Tripp, B. & Eliasmith, C. (2007). Neural populations can induce reliable postsynaptic currents without observable spike rate changes or precise spike timing. *Cereb. Cortex*, 17(8), 1830–1840. (p 17).

Trousdale, J., Hu, Y., Shea-Brown, E., & Josic, K. (2012). Impact of network structure and cellular response on spike time correlations. *PLoS Comput. Biol.*, 8(3), e1002408. (p 89).

Tsodyks, M., Pawelzik, K., & Markram, H. (1998). Neural networks with dynamic synapses. *Neural Comput.*, 10, 821–835. (p 38).

Tsodyks, M., Uziel, A., & Markram, H. (2000). Synchrony generation in recurrent networks with frequency-dependent synapses. *J. Neurosci.*, 20, RC1 (1–5). (p 38).

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley. (pp 27, 167).

Turrigiano, G. G. (2008). The self-tuning neuron: synaptic scaling of excitatory synapses. *Cell*, 135(3), 422–435. (p 127).

van Albada, S. J., Helias, M., & Diesmann, M. (2015). Scalability of asynchronous networks is limited by one-to-one mapping between effective connectivity and correlations. *PLoS Comput. Biol.*, 11(9), e1004490. (pp 30, 89).

van Albada, S. J., Kunkel, S., Morrison, A., & Diesmann, M. (2014). Integrating brain structure and dynamics on supercomputers. In L. Grandinetti, T. Lippert, & N. Petkov (Eds.), *Brain-Inspired Computing* (pp. 22–32). Springer. (p 123).

Van Essen, D. C. (2002). Windows on the brain: the emerging role of atlases and databases in neuroscience. *Curr. Opin. Neurobiol.*, 12(5), 574–579. (pp 74, 98, 159).

Van Essen, D. C. (2012). Cortical cartography and caret software. *NeuroImage*, 62(2), 757–764. (p 158).

Van Essen, D. C., Drury, H. A., Dickson, J., Harwell, J., Hanlon, D., & Anderson, C. H. (2001). An integrated software suite for surface-based analyses of cerebral cortex. *Journal of the American Medical Informatics Association*, 8(5), 443–459. (pp 71, 74, 98, 153, 159).

van Kerkoerle, T., Self, M. W., Dagnino, B., Gariel-Mathis, M.-A., Poort, J., van der Togt, C., & Roelfsema, P. R. (2014). Alpha and gamma oscillations characterize feedback and feedforward processing in monkey visual cortex. *Proc. Natl. Acad. Sci. USA*, 111(40), 14332–14341. (pp 18, 126).

van Vreeswijk, C. & Sompolinsky, H. (1996). Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274, 1724–1726. (p 89).

Vandervoorde, D. & Josuttis, N. (2003). *Templates: the complete guide*. Boston MA, USA: Addison Wesley, 1st edition. (p 51).

Vincent, J., Patel, G., Fox, M., Snyder, A., Baker, J., Van Essen, D., Zempel, J., Snyder, L., Corbetta, M., & Raichle, M. (2007). Intrinsic functional architecture in the anaesthetized monkey brain. *Nature*, 447(7140), 83–86. (pp 18, 18, 130).

Voges, N. & Perrinet, L. U. (2012). Complex dynamics in recurrent cortical networks based on spatially realistic connectivities. *Front. Comput. Neurosci.*, 6, 41. (p 3).

Voges, N., Schüz, A., Aertsen, A., & Rotter, S. (2010). A modeler's view on the spatial structure of intrinsic horizontal connectivity in the neocortex. *Prog. Neurobiol.*, 92(3), 277–292. (p 130).

Voorhees, E. M. (1986). Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. *Information Processing & Management*, 22(6), 465–476. (pp 104, 106).

Wagstyl, K., Ronan, L., Goodyer, I. M., & Fletcher, P. C. (2015). Cortical thickness gradients in structural hierarchies. *NeuroImage*, 111, 241–250. (p 125).

Waters, J. & Helmchen, F. (2006). Background synaptic activity is sparse in neocortex. *J. Neurosci.*, 26(32), 8267–8277. (p 26).

Webster, M., Ungerleider, L., & Bachevalier, J. (1991). Connections of inferior temporal areas TE and TEO with medial temporal-lobe structures in infant and adult monkeys. *J. Neurosci.*, 11(4), 1095–1116. (p 82).

Webster, M. J., Bachevalier, J., & Ungerleider, L. G. (1994). Connections of inferior temporal areas TEO and TE with parietal and frontal cortex in macaque monkeys. *Cereb. Cortex*, 4(5), 470–483. (p 82).

Wedeen, V. J., Wang, R., Schmahmann, J. D., Benner, T., Tseng, W., Dai, G., Pandya, D., Hagmann, P., D'Arceuil, H., & de Crespigny, A. J. (2008). Diffusion spectrum magnetic resonance imaging (dsi) tractography of crossing fibers. *NeuroImage*, 41(4), 1267–1277. (p 89).

Weisstein, E. W. (2005). Beta binomial distribution. *From MathWorld–A Wolfram Web Resource.* (p 80).

Wong, K.-F. & Wang, X.-J. (2006). A recurrent network mechanism of time integration in perceptual decisions. *J. Neurosci.*, 26(4), 1314–1328. (p 92, 92).

Yonezawa, A., Watanabe, T., Yokokawa, M., Sato, M., & Hirao, K. (2011). Advanced institute for computational science (aics): Japanese national high-performance computing research institute and its 10-petaflops supercomputer "k". In *State of the Practice Reports*, SC '11 (pp. 13:1–13:8). New York, NY, USA: ACM. (p 47).

Zilles, K., Palomero-Gallagher, N., & Schleicher, A. (2004). Transmitter receptors and functional anatomy of the cerebral cortex. *Journal of Anatomy*, 205(6), 417–432. (p 128).

Zito, K. & Scheuss, V. (2009). Nmda receptor function and physiological modulation. *Encyclopedia of neuroscience*, 6, 1157–1164. (p 131).

Author's list of publications

1. **Schmidt M**, Bakker R, Shen K, Bezgin G, Hilgetag C-C, Diesmann M, and van Albada SJ (2015).
   Full-density multi-scale account of structure and dynamics of macaque visual cortex.
   arXiv:1511.09364
   Results presented in Chap. 3 and Chap. 5.

2. Schuecker J*, **Schmidt M** *, van Albada SJ, Diesmann M, and Helias M. (2015)
   Fundamental activity constraints lead to specific interpretations of the connectome
   arXiv:1509.03162
   * Both authors contributed equally.
   Results presented in Chap. 4.

3. Kunkel S, **Schmidt M**, Eppler JM, Plesser HE, Masumoto G, Igarashi J, Ishii S, Fukai T, Morrison A, Diesmann M and Helias M (2014)
   Spiking network simulation code for petascale computers.
   Frontiers in Neuroinformatics, 8:78
   Results presented in Chap. 2.

Danksagung

Diese Arbeit wäre ohne die Unterstützung vieler Menschen, denen ich hier explizit danken möchte, nicht möglich gewesen.
Zuerst möchte ich Markus Diesmann danken, mein Doktorvater, der mir in den fast vier Jahren mit Rat und Tat zur Seite stand und mir viele Dinge zugetraut hat, an denen ich gewachsen bin.
Ich danke Prof. Offenhäusser für seine Bereitschaft, meine Arbeit als Zweitgutachter zu beurteilen.
Mein größter Dank gebührt Sacha van Albada. Sie hat als meine direkte Betreuerin meine Arbeit mit ihrem enormen Fachwissen und ihrer wissenschaftlichen Gründlichkeit entscheidend geprägt.
Ich danke Moritz Helias dafür, dass er immer hilfsbereit meine Fragen beantwortet hat und die einzelnen Projekte dieser Arbeit durch seine wissenschaftliche Intuition vorangebracht hat.

I am grateful to Kelly Shen and Gleb Bezgin who contributed a valuable part to this thesis by providing the experimental resting-state data. The same holds for Helen Barbas, Claus Hilgetag and Sarah Beul who provided structural data which was essential for the construction of the network model. Rembrandt Bakker contributed immensely to this work by explaining all details of the CoCoMac database and helping with the interpretation of connectivity and anatomical data.
For the development of NEST 4g, we gratefully acknowledge Ryutaro Himeno for advising us from the start on the research programs developing and using the K computer, Mitsuhisa Sato for hosting our activities at RIKEN AICS in the later phase of the project, and Jakob Jordan for help in collecting the simulation data.

Ich danke allen Kollegen am INM-6; ich kenne kein anderes Institut, das derart von Kooperation, Hilfsbereitschaft und offener Diskussionskultur geprägt ist.
Dabei möchte ich besonders Jannis danken, mit dem mir die Zusammenarbeit immer großen Spaß bereitet hat. Außerdem gilt mein Dank Jakob, Lyuba und David, die immer ein offenes Ohr für mich hatten und mich fachlich und mental unterstützt haben. Ich danke auch Emiliano, mit dem ich in den letzten Monaten vor der Abgabe dieser Arbeit in einer Schicksalsgemeinschaft einige Klippen umschifft habe. Susanne hat mir bei meiner Arbeit an NEST gerade in

der Anfangszeit enorm geholfen und mir geduldig alle Fragen beantwortet, vielen Dank dafür. Ebenso danke ich Bernd, Jochen und Tiziano für ihre Hilfsbereitschaft in allen Fragen zu diversen Hardware- und Softwareproblemen.
Für die vielen aufreibenden, aber auch entspannenden Spiele in der Kickerküche danke ich allen Spielern am Institut.
Ein besonderer Dank geht an hambach37; Du warst mir ein treuer Begleiter.
Mika, für Deine Unterstützung aus der Nähe und der Ferne: ありがとうございました。
Zu guter Letzt möchte ich meinen Freunden und meiner Familie für ihre Unterstützung in dieser nicht immer einfachen Zeit danken.

Funding

This work was supported by the Helmholtz Portfolio Supercomputing and Modeling for the Human Brain (SMHB), European Union (BrainScaleS, grant 269921 and Human Brain Project, grant 604102), the Jülich Aachen Research Alliance (JARA), Helmholtz Alliance on Systems Biology, the Initiative and Networking Fund of the Helmholtz Association, the Helmholtz young investigator group VH-NG-1028, the Next-Generation Supercomputer Project of MEXT, Research Council of Norway Grant 178892/V30 (eNeuro) and access to NOTUR supercomputing facilities.

Computing time on JUQUEEN was granted by the JARA-HPC Vergabegremium and provided on the JARA-HPC Partition part of the supercomputer JUQUEEN (Jülich Supercomputing Centre, 2015) at Forschungszentrum Jülich (VSR computation time grant JINB33). Computing time on the K computer was provided through early access in the framework of the co-development program, project hp130120 of the General Use Category (2013), the Strategic Program (Neural Computation Unit,OIST), and MEXT SPIRE Supercomputational Life Science.